CLUBE

Abril/85

N.º 31

Coordenadores: ANA PAULA • ALEXANDRE SOUSA • J. MAGALHAES • AV.ª BOAVISTA, 832-2.º T. • TELEF. 65127 • 4100 PORTO

NESTE NÚMERO

INTRODUÇÃO AO CÓDIGO MÁQUINA 1
TIMEX 2068 VAMOS VER O QUE É 6
FDD — COMO PREPARAR UMA DISKETTE/SISTEMA OPERATIVO TIMEX 6

Novos Programas:

— MICROCOPI — PROGRAMA DE CÓPIA 8
— FAÇA O SEU PRÓPRIO COPIADOR 10
— TESTE VOCACIONAL (Cont.) 11
— FICHEIRO DE CASSETES 12
— PROGRAMA BIORRITMO 16

RESPOSTA AO DESAFIO 17
PORMENORES 18
NOVOS PROGRAMAS 19

No interior:

Folheto Mercado Z80

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Ramos dos Santos & C.ª, Lda./Porto

Tiragem: 500 exemplares, Abril 1985

INTRODUÇÃO AO CÓDIGO MÁQUINA

(Cont. dos números anteriores)

Autor: FERNANDO PRECES
SACAVÉM

ENSAIO 3 : SPECTRUM

Como imprimir Strings

```
         ORG     40000
PARTM:   DEFB    17    ; comando PAPER
         DEFB    0
         DEFB    16    ; comando INK
         DEFB    5
         DEFB    22    ; comando AT
         DEFB    10
         DEFB    8
STRING:  DEFB    70    ; mensagem STRING
         DEFB    69
         DEFB    76
         DEFB    73
         DEFB    90
         DEFB    32
         DEFB    65
         DEFB    78
         DEFB    79
         DEFB    32
         DEFB    78
         DEFB    79
         DEFB    86
         DEFB    79
TOTAL:   DEFW    21    ; extensão total
                       — STRING+parâmetros
START:   LD A    2
         CALL    5633  ; abre canal S (Écran)
         LD B    24
         CALL    3652  ; CLS às 24 linhas
         LD DE PARTM   ; aponta início dos parâmetros
         LD BC (TOTAL)
         CALL    8252  ; imprime a STRING
         RET
```

Utilizar o assemblador (descrito em suplemento) para introduzir na memória os parâmetros e conteúdo duma String é pouco prático, visto este assemblador exigir a entrada de dados separadamente, um a um.

O leitor deve utilizar o assemblador apenas para assemblar as mnemónicas e introduzir os restantes dados, tais como tabelas numéricas, texto, etc., por outros processos.

Quando se prepara código máquina e não possuímos um assemblador profissional, aonde tudo é fácil, podemos utilizar o Basic para introduzir esses dados, ou por uma linha DATA no caso dos parâmetros, ou por uma linha REM no caso do texto, que depois facilmente colocamos no local pretendido da memória. (Veja, para exemplo, pequenos monitores — Capítulo 1).

Após todos os dados estarem armazenados na REM 1, pode gravá-los com SAVE «Data» Code 23760, n e chamá-los depois para a memória com LOAD "" Code **pretendido**.

Uma vez concluída a rotina em C/M, pode ensaiá-la com:

100 RANDOMIZE USR **START**

Em que START é o nome atribuído no ensaio ao endereço de arranque que o leitor conhecerá logo após a assemblagem.

ENSAIO 4: ZX81

Como imprimir Strings

A rotina que vamos utilizar é parte integrante da rotina de comando PRINT do programa monitor. Ela é chamada sempre que uma String de caracteres tem de ser escrita.

Os parâmetros para o seu manuseamento são dois. O endereço de arranque e o número total de caracteres da String.

```
        ORG 16444
START:  LD DE 16514   ; endereço da String
        LD BC 29      ; extensão da String
        CALL 2923     ; ROM — comando PRINT String
        RET
```

```
10 REM ESTA STRING TEM 29 CARACTERES
20 SLAW
30 PRINT AT 18,0;
40 RAND USR 16444

   RUN...
```

Subgrupo D: as instruções RST

RST é uma abreviatura de RESTART que nos dá uma ideia de recomeço ou rearranque.

As rotinas RST são na realidade especiais. Repare-se que o fabricante da ROM as distingue de todas as outras, porque utiliza apenas **um byte** para as chamar.

Apesar de serem instruções muito semelhantes à instrução CALL incondicional, não requerem endereço de referência. Cada uma delas (oito ao todo), comanda uma importante rotina monitora, a detalhar no Capítulo 3.

Mnemónicas	Códigos	Nomes das Rotinas
RST 0	199	The start
RST 8	207	The error restart
RST 16	215	Print a charactere restart
RST 24	223	The collect charactere restart
RST 32	231	The collect next charactere restart
RST 40	239	The calculator restart
RST 48	247	The make be spaces restart
RST 56	255	The mascable interrupt

Antes de efectivar o Salto para qualquer destas rotinas, com a excepção da primeira (RST 0), o Z80 envia uma cópia do endereço de PC para o Stack, assegurando o retorno à sequência interrompida.

Exemplos de aplicação:

RST 0

Alguns programas de jogos da última geração, aparecem condicionados à introdução pelo teclado de um ou mais códigos chave, com variantes entre números, letras e cores. Quando os códigos introduzidos não são os correctos, o programa é limpo da memória. Darei aqui um exemplo como pode ser conseguida essa operação.
Existem, em código máquina, vários processos para pesquisar o teclado e devolver a informação sobre a tecla premida. O mais simples para este exemplo, envolve o teste a 2 variáveis de sistema, a KSTATE (endereço 23557) e a LAST K (endereço 23560).
A rotina monitora de pesquisa ao teclado, memoriza nos endereços destas variáveis, 50 vezes por segundo, se houve tecla premida e qual.

```
        KSTATE   EQU 23557
        LAST K : EQU 23560
START:  LD A , (KSTATE)
        SUB 5
        JR NZ, START   ;  não tecla premida
        LD A   (LAST K);  última premida
```

Segue-se a memorização da tecla em HL, que está apontado para determinado endereço da memória, a pesquisa de outro código, nova memorização, etc. Após a entrada de todos os elementos código, compara-se e...

```
JR NZ, FIM          ;  erro no código
JP endereço jogo    ;  OK código certo
FIM: RST 0          ;  a memória é limpa
```

A pesquisa do teclado foi aqui apenas apresentada como exemplo, porque mais adiante vai ser objecto de uma conversa bem mais detalhada.
A instrução RST 0 provoca um NEW total ao sistema, tal como tivéssemos interrompido momentaneamente a corrente à máquina.

ENSAIO 1: ZX81
Introdução de texto no écran através da rotina RST 16

O texto pode ser introduzido pelo Basic na primeira REM e depois transportado para a zona de memória pretendida.
A separação entre cada linha do texto é conseguida pela introdução o símbolo "*" (código 23).
Por cada símbolo "*" a mais, obterá uma linha em branco.
O final do texto é indicado pelo símbolo "$", (código 13).

Exemplo da formação da REM:

```
1 REM MENU ** PARTE 1 — ESTRATÉGIA *
  PARTE 2 — COMBATE ** TECLAS DE COMANDO:
  ** 0... POSIÇÃO NO ÉCRAN * 1, 2, 3 ...
  TIPO DE PEÃO * Z... FIXAÇÃO * etc... $
```

Assembler da rotina:

```
        TEXTO: EQU 30000
        ORG 30500
START:  LD HL TEXTO
LOOP:   LD A (HL)
        CP 23           ;  caractere "*" ?
        JR Z A1
        CP 13           ;  caractere "$" ?
        JR Z B1
        RST 16
        INC HL
        JR LOOP
A1:     INC HL
        LD A 118        ;  NEW LINE
        RST 16
        JR LOOP
B1:     RET
```

Códigos da rotina:

33, 48, 117, 126, 254, 23, 40, 8, 254, 13, 40, 10, 215, 35, 24, 243, 35, 62, 118, 215, 24, 237, 201.

```
100 PRINT AT 3,0;
110 RAND USR 30500
RUN...
```

O comando PRINT AT também pode ser substituído por uma rotina da ROM. (Veja introdução ao C/M. ZX81 — Capítulo 1).

ENSAIO 2: SPECTRUM

Introdução de texto no écran através da rotina RST 16

Imprima na REM 1, 5 frases escritas em cores diferentes de tinta e papel, usando os comandos directos de cor.
Informações tais como inverso de vídeo, intensidade luminosa, sobreposição de caracteres e intermitência, podem também ser indicados usando os comandos directos respectivos.
Como separador entre cada frase, utilize o caracter "*" e repita-o tantas vezes, quantas linhas quiser de intervalo entre estas. A finalizar o texto, imprima o caracter "#".
Passe os bytes da REM para o endereço 60000, pelo processo já recomendado.

Rotina a assemblar com RAMTOP em 59000

```
        TEXTO: EQU 60000
        ORG 60500
START:  LD HL TEXTO
LOOP:   LD A (HL)
        CP 42           ;  caracter "*" ?
        JR Z A1
        CP 35           ;  caracter "#" ?
        JR Z B1
        RST 16
        INC HL
        JR LOOP
A1:     INC HL
        LD A 13         ;  NEW LINE
        RST 16
        JR LOOP
B1:     RET
```

Ao assemblar esta rotina vai reparar que na 1.ª passagem o assemblador marcará com * (inverso) as instruções JR Z A1 e JR Z B1. No final da assemblagem anote os endereços de

A1 e B1, faça um STOP ao programa, liste as citadas instruções e substitua A1 e B1 pelos respectivos endereços. Execute uma segunda passagem ao programa com GOTO 1020 e se não houver erro grave os respectivos códigos. Finda a assemblagem, junta-se os bytes do texto com os bytes da rotina. O texto é chamado com o endereço base em 60000 e a rotina em 60500.

Para o ensaio, introduza o seguinte Basic:

```
90 PRINT AT 3,0;
100 RANDOMIZE USR 60500
```

Vão surgir no écran todas as frases anteriormente impressas na REM 1, com os seus atributos.

Subgrupo E — As instruções RET

As instruções RET incondicional e todas as outras RETs condicionadas, fazem parte deste conjunto de instruções. A principal utilização destas instruções, para além do retorno ao Basic, já nosso conhecido, é a de **marcar** o final duma subrotina e proporcionar o regresso ao programa principal. O Z80 ao ler uma instrução RET, extrai do Stack o endereço de retorno que coloca no registro PC, e incrementa 2 vezes o registro SP.

Mnemónicas	Códigos	Comentários
RET	201	Retorno incondicional
RET C	216	» se carry flag = 1
RET NC	208	» se carry flag = 0
RET Z	200	» se zero flag = 1
RET NZ	192	» se zero flag = 0
RET M	248	» se sinal flag = 1
RET P	240	» se sinal flag = 0
RET PE	232	» se O/P flag = 1
RET PO	224	» se O/P flag = 0

É importante ter em conta que o endereço de retorno extraído do Stack por uma destas instruções RET, não teve de ser necessariamente lá colocado pela correspondente instrução CALL.

O exemplo que darei a seguir, extraído do programa monitor do ZX81, mostra-nos como pode ser utilizada uma tábua de endereços, que vai proporcionar a selecção do endereço de retorno após a execução da subrotina.

Na 'Basic line scanning routine' (Rotina que pesquisa os elementos que compõem a linha Basic em execução), os diferentes comandos Basic estão agrupados em sete conjuntos de classe. Assim, o comando da linha pesquisada é determinado e um salto é efectuado para a respectiva rotina de classe, cujos códigos de referência vão de 0 a 6.

O código seleccionado é carregado no registro C.

Endereços	Mnemónicas	Comentários
3333	LD HL, 3350	Endereço base da tábua
3336	LD B, 0	Clear B
3338	ADD HL, BC	Soma o código de classe
3339	LD C, HL	Põe conteúdo em C
3340	ADD HL, BC	Forma o endereço
3341	PUSH HL	Coloca-o no Stack
3342	RST 24	Lê o caracter seguinte
3343	RET	Salta para o endereço do comando

Tábua de endereços das classes de comando

Endereço	Conteúdo	Endereço obtido	Classe
3350	23	3373	0
3351	37	3388	1
3352	83	3435	2
3353	15	3368	3
3354	107	3461	4
3355	19	3374	5
3356	118	3474	6

ENSAIO 1: ZX81 E SPECTRUM

Rotina para transcrever no écran pequenas frases, tais como PRIMA UMA TECLA, FIM DO JOGO, etc., ou ainda símbolos e caracteres formados pelo utilizador, para o caso do Spectrum.

O texto é colocado no endereço Q1 e cada frase separada por "*".

```
START: LD HL Q1
 LOOP: LD A (HL)
       CP "*"
       RET Z
       RST 16
       INC HL
       JR LOOP
```

GRUPO 13 — As instruções de rotação

O Z80 tem um grande número de instruções para a rotação de bits.

Quando bem aproveitadas, essas instruções podem ser muito úteis. No entanto, são pouco utilizadas pela maioria dos programadores, talvez pela dificuldade de as distinguir devido aos seus nomes serem difíceis de fixar e pelo facto da acção se desenrolar ao nível de bits.

Na minha opinião, não as devemos desprezar pois a sua actuação sobre representações gráficas têm um efeito surpreendente, isto para além de outras aplicações não menos importantes.

Para simplicidade no seu estudo, vamos dividi-las em 3 subgrupos.

Subgrupo A — As instruções de **código único** que afectam apenas o carry flag e que são destinadas ao registro A.

Mnemónicas	Códigos
RLC	7
RRC	15
RL	23
RR	31

Subgrupo B — As instruções de **código duplo** que afectam todos os flags.

Estas mnemónicas não afectam os flags de uma maneira diferente do já relatado com outros tipos de instruções. Contudo, tentando desmistificar o rótulo de «instruções muito difíceis de aplicação», farei aqui um apanhado do comportamento de cada flag, após uma instrução de rotação.

Carry flag — Toma o valor 1 quando o seu conteúdo for 1.

Zero flag — Toma o valor 1 quando o byte resultante for 0.

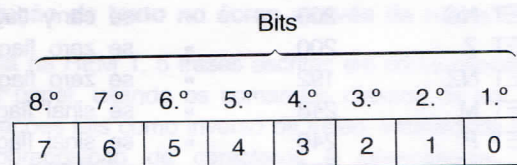
Sign flag — Toma o valor 1 quando o bit de sinal do resultado for 0.

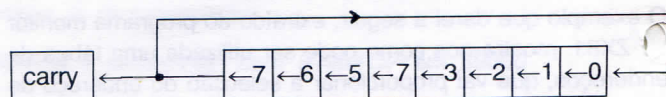
O/P flag — Toma o valor 1 se a paridade do resultado for par.

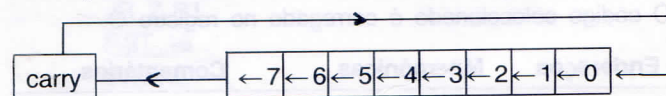
Mnemónicas		**Códigos**
RLC	A	203, 7
RLC	B	203, 0
RLC	C	203, 1
RLC	D	203, 2
RLC	E	203, 3
RLC	H	203, 4
RLC	L	203, 5
RLC	(HL)	203, 6
RLC	(IX + d)	221, 203, + d, 6
RLC	(IY + d)	253, 203, + d, 6
RL	A	203, 23
RL	B	203, 16
RL	C	203, 17
RL	D	203, 18
RL	E	203, 19
RL	H	203, 20
RL	L	203, 21
RL	(HL)	203, 22
RL	(IX + d)	221, 203, + d, 22
RL	(IY + d)	253, 203, + d, 22
SLA	A	203, 39
SLA	B	203, 32
SLA	C	203, 33
SLA	D	203, 34
SLA	E	203, 35
SLA	H	203, 36
SLA	L	203, 37
SLA	(HL)	203, 38
SLA	(IX + d)	221, 203, + d, 38
SLA	(IY + d)	253, 203, + d, 38
RRC	A	203, 15
RRC	B	203, 8
RRC	C	203, 9
RRC	D	203, 10
RRC	E	203, 11
RRC	H	203, 12
RRC	L	203, 13
RRC	(HL)	203, 14
RRC	(IX + d)	221, 203, + d, 14
RRC	(IY + d)	253, 203, + d, 14
RR	A	203, 31
RR	B	203, 24
RR	C	203, 25
RR	D	203, 26
RR	E	203, 27
RR	H	203, 28
RR	L	203, 29
RR	(HL)	203, 30
RR	(IX + d)	221, 203, + d, 30
RR	(IY + d)	253, 203, + d, 30
SRA	A	203, 47
SRA	B	203, 40
SRA	C	203, 41
SRA	D	203, 42
SRA	E	203, 43
SRA	H	203, 44
SRA	L	203, 45
SRA	(HL)	203, 46
SRA	(IX + d)	221, 203, + d, 46
SRA	(IY + d)	253, 203, + d, 46
SRL	A	203, 63
SRL	B	203, 56
SRL	C	203, 57
SRL	D	203, 58
SRL	E	203, 59
SRL	H	203, 60
SRL	L	203, 61
SRL	(HL)	203, 62
SRL	(IX + d)	221, 203, + d, 62
SRL	(IY + d)	253, 203, + d, 62

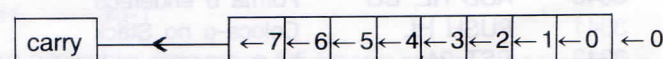
Quando uma instrução de rotação de bits é executada, todos os bits pertencentes ao registro indicado, se deslocam uma casa. A direcção da deslocação depende do tipo de instrução.

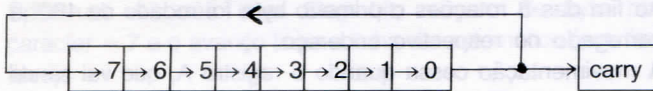
Vamos considerar os 8 bits dum registro e referenciá-los devidamente.

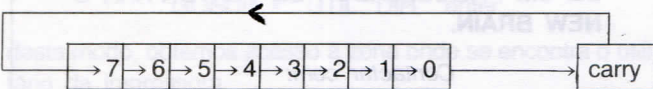
Bits							
8.º	7.º	6.º	5.º	4.º	3.º	2.º	1.º
7	6	5	4	3	2	1	0

Assim, o 1.º bit dum registro é referenciado como **bit 0** e sucessivamente até ao 8.º bit referenciado como **bit 7.**

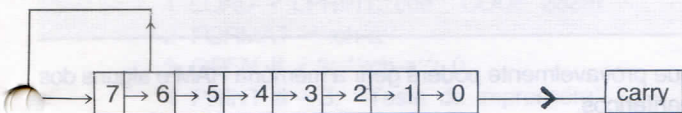
Instruções do tipo RLC — Rotação à esquerda com carry.

O carry flag toma o valor do bit 7, todos os bits do registro se deslocam uma posição à esquerda e o bit 0 tomará o valor que pertencia ao bit 7.

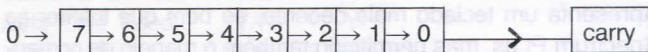
Instruções do tipo RL — Rotação à esquerda.

Bit 7 vai para o carry. Carry para o bit 0.

Instruções do tipo SLA — Transferência à esquerda.

O bit 0 é colocado **a zero.** O bit 7 vai para o carry.

Instruções do tipo RRC — Rotação à direita, com carry.

O bit 0 vai para o carry e toma a posição do bit 7.

Instruções do tipo RR — Rotação à direita.

O bit 0 vai para o carry. Carry para o bit 7.

Instruções do tipo SRA — Transferência à direita.

O bit 0 vai para o carry e o bit 7 **não altera.**

Instruções do tipo SRL — Transferência à direita.

Carry recebe o bit 0. O bit 7 é colocado a zero.

Um exemplo extraído do monitor do ZX81, mostra-nos a utilização duma instrução RL E na gravação dum programa. Na rotina de comando SAVE, o nome dum programa, o programa e variáveis, são passados para a cassete da seguinte forma:
Cada byte do programa é carregado no registro E que é rodado a esquerda após o carry flag ser colocado em 1. Ao ser colocado o **mono bit** (o bit do carry) no valor 1, efectua-se uma **marca** indicadora do fim de um byte.
Na primeira rotação o bit 7 vai para o carry, na segunda o bit 6 e assim sucessivamente até entrar o bit 0.
Por cada rotação, o carry é testado e o seu valor enviado para a cassete, sendo depois **limpo** antes da rotação seguinte. No final dos 8 ciclos, é efectuada outra rotação para testar o carry. Se este contiver a **marca** o registro E recebe o byte seguinte para gravar.

Endereços	Mnemónicas	Comentários
798	LD E (HL)	Chama um byte
799	SCF	Carry em 1 para a marca
800	RL E	Rotação de E à esquerda
802	RET Z	Retorno após 8 ciclos
. . .		
827	AND A	Limpa carry
828	DJNZ 827	Linha de atraso que determina a velocidade de gravação
830	JR 800	Repete o ciclo

ENSAIO 1 (ZX81 E SPECTRUM)

Multiplicando com instruções de rotação

Carregue o registro BC com um número entre 1 e 8190 (é necessário que o produto não exceda 65535, valor máximo de acumulação para registro par).

```
LD BC NN
SLA C        203, 33
RL B         203, 16
SLA C        203, 33
RL B         203, 16
RL B         203, 16
RET
                  PRINT USR...
```

O número introduzido é multiplicado por 2 ↑ 3 e o resultado transportado por BC para o Basic.
Utilizando as duas instruções similares de rotação à direita pode efectuar pelo mesmo processo a operação inversa (a divisão).

ENSAIO 2 (ZX81 E SPECTRUM)

```
LD     BC     NN   (por exemplo 65520)
SRL    B           203, 56
RR     C           203, 25
SRL    B           203, 56
RR     C           203, 25
SRL    B           203, 56
RR     C           203, 25
RET
                  PRINT USR...
```

O número introduzido é dividido por 2 ↑ 3 e o resultado transportado por BC para o Basic

ENSAIO 3 (SPECTRUM)

Rotação de 180° de um conjunto de caracteres gráficos

Vamos supor que temos 2 caracteres gráficos representando a figura de um boneco a andar para a esquerda e que a dada altura o quer rodar para o pôr a andar para a direita.

Exemplo:

```
START:   ORG 23300
         LD HL (23675)   ; variável UDG
         LD A 16         ; para 2 caracteres gráficos
NEXT B   LD B 8          ; byte seguinte
```

```
NEXT P  RR (HL)        ; pixel seguinte
        RL C
        DJNZ NEXT P
        LD (HL) C
        INC HL
        DEC A
        JR NZ NEXT B
        RET

        RANDOMIZE USR 23300
```

O registro HL aponta o endereço do primeiro caracter gráfico (UDG), o registro A é carregado com o número de caracteres a modificar (8 bytes * 2 = 16), o registro B recebe o número 8 (8 bits por byte) que vai usar para a contagem.
O primeiro byte apontado por HL é rodado para a direita, sendo o primeiro bit introduzido no carry flag que o passa para o registro C que vai rodar à esquerda.
Ao fim das 8 rotações o primeiro byte foi rodado de 180º e carregado no respectivo endereço.
A movimentação cessa quando o registro A, que vai sendo decrementado, chegar a zero.

TIMEX 2068 vamos ver o que é!

Anunciado há meses (para não dizer há dois anos) a Timex parece finalmente decidida a lançar uma máquina que embora semelhante (ou mais ou menos igual), não dependa da Sinclair.
É interessante não esquecer que todas as ULAs (circuito integrado responsável pelas entradas/saídas do Spectrum), são fornecidas pela SINCLAIR o que obriga a TIMEX a uma certa ginástica de fornecimentos mais ou menos esquisitos, de tal modo que o Spectrum aparece no Norte de Portugal, a um preço que varia entre 19 e 28 contos.
Acontece que alguns felizardos já observaram a nova máquina, denominada TIMEX COMPUTER 2068 e de acordo com a publicidade do produtor, a tónica que tenta empurrar a venda desta máquina, será a da capacidade de memória . . . 72 Kbytes de RAM.
Claro que isto, visto sob o ponto de vista do utilizador com alguma experiência, não pode ser apresentado desta forma.
É dos livros, que um microprocessador de 8 bits (tal como o Z80), não pode trabalhar com memória superior a 64 Kbytes... a não ser que o Sistema Operativo seja de tal modo potente que consiga ele próprio Sistema Operativo, gerir a memória.
Falam-nos de utilização de uma ROM externa de 56 Kbytes, que provavelmente poderá gerir a memória RAM e alguns dos periféricos.

A máquina pode trabalhar em dois modos distintos, ou seja, como um Spectrum vulgar e mantém todas as características do nosso velho conhecido, ou pode ser comutado para 24 linhas x 64 colunas, o que se aproxima dos modelos mais ou menos «profissionais».

Apresenta um teclado mais decente, se bem que inferior ao Spectrum PLus, mas permitindo também o manejo de comandos com uma só tecla, sem ter de usar a técnica da «aranha», como quando queremos usar a função INT no Spectrum (lembram-se?).

Tem saídas distintas para Monitor Monocromático (fósforo verde, por exemplo) e TV cor ou p/b.
Pode usar dois Joysticks, um gravador, e possui a saída habitual para acoplar o Interface 1 RS232 ou o controlador das Diskettes (FDD).

Com toda a sinceridade, não nos parece ter detectado uma grande curiosidade à volta deste anúncio, será que os aficionados duvidam que este projecto tenha sido finalmente concluído?

FDD — Como preparar uma Diskette/Sistema Operativo Timex

ALEXANDRE SOUSA

Surgiram recentemente os drives para leitura e reprodução de programas (Sinclair/Spectrum) e que foram lançados pela TIMEX. O sistema é denominado TOS-Timex Operating System — e vai criar o meio ambiente no qual serão entendidos os comandos e instruções que permitem usar diskettes como memória externa e valiosa.
As diskettes possuem o formato de 3 polegadas e capacidade para 160 Kbytes de cada lado, ou seja, 320 k em ambos os lados.
O Hardware classificado como FDD — Floppy Disk Drive — é de concepção japonesa e adaptado pela Timex para funcionamento conjunto com o Spectrum 48 K e com o recém-criado Timex 2068.
O Hardware é composto por um Drive-dispositivo que grava e reproduz as diskettes; por uma fonte de alimentação e por um controlador que possui também uma saída tipo RS232, denominada CANAL «ch-a».
Independentemente de muitos problemas que se levantam sobre a manipulação dos ficheiros, sob o ponto de vista da utilização das diskettes, vamos apresentar hoje o procedimento completo, relativo à preparação de uma diskette que permita obter uma listagem do programa numa impressora de 80(132) colunas, dada a dificuldade inicial de que esta acção se revestiu.
A impressora que usamos é do tipo STAR DELTA 10, que possui as vantagens de ter incorporados os dois tipos de comunicação — Série e Paralelo — e possuir uma velocidade de 160 caracteres por segundo; ser bidireccional e permitir usar papel do tipo formulário e normal.
O Interface da impressora foi previamente preparado para uma velocidade de transmissão de 1200 Banda — não é

aconselhável usar elevada velocidade de transmissão com o Spectrum —; não usar Stop bit; número de byte por caracter = 7 e o avanço linha a linha ser automático (aut. line feed).
Consultar o manual da impressora, para verificar se existem as mesmas condições, se a impressora não é STAR.
Em seguida, ligam o FDD e com a diskette de origem, denominada «DEMO» disk, chamamos o Directório onde se encontra o utilitário da impressora.

```
GOSUB * ''UTIL. DIR'' enter
```

deste modo, obtemos acesso à zona onde se encontra o utilitário da impressora

```
LOAD * ''LPRINT''
```

Observamos que possui apenas uma instrução na linha 9.
Vamos modificar de modo a escrever:

```
1 LOAD *''LPRINT. cod'' CODE 23297
2 FORMAT *'':ch-a''
3 OPEN # * 3; '':ch-a''; 0
4 PRINT # * 3; ''Teste da impressora''
5 Close # * 3
```

Ligar a impressora, preparada com papel, etc.
Vamos fazer correr o propgrama e iremos em seguida responder às seguintes questões:

```
TEXT or BYTES      T/B)  ?  T
Auto Line Feed     (Y/N)  ?  Y
XON/X OFF          (Y/N)  ?  N
INPUT with wait    (Y/N)  ?  N
BAUDE RATE         (A/P)  ?  I     (nosso caso 1200)
PARITY (E = Even, O = Odd, N = None)  ?  N
STOP BITS (A = 1, B = 1.5, C = 2)  ?  Enter
BITS/Char (A = 5, B = 6, C = 7, D = 8)  ?  C
```

Se tudo estiver de acordo com as nossas respostas, iremos observar a saída para a impressora, da frase que está contida na linha 4 do nosso programa.
Quando obtivermos a saída na impressora, iremos preparar uma diskette, que vai armazenar os nossos programas de modo a não ser necessário, responder a toda aquela série de perguntas, sempre que precisar de usar a impressora.
Pegue numa diskette virgem, ou que possa formatar (não esquecer que este comando, elimina tudo o que está gravado na diskette!).
Ainda com a diskette «DEMO», use o comando directo:

```
FORMAT * ''A'' TO ''DISK 1''
```

Nós usamos ''disk 1'' no nosso exemplo. Use o nome que entender para a diskette que quer formatar.
À pergunta FORMAT DISK IN DRIVE A (Y/N)? responda Y
Retire a diskette e introduza a diskette que vai usar com os seus programas.
Quando a diskette estiver formatada, volte a introduzir a diskette ''DEMO'', volte a selecionar: GOSUB * ''util. dir''

```
LOAD * ''LPrint''
```

Escreva sem número de linha:

```
LOAD * ''LPrint. cod'' CODE 23297
:INPUT ''disk dos programas''; Y$
:SAVE * ''LPrint. cod'' CODE 23297, 76 Enter
```

Quando observar no écran o pedido da diskette do programa, introduza a sua nova diskette e use o comando Enter.
Agora no seu primeiro programa, a linha inicial deve ser:

```
LOAD * ''LPrint. cod'' CODE 23297
RANDOMIZE USR 23297
OPEN #*3; '':ch—a'';0
```

Desta forma sempre que use a impressora, terá o canal 3 atribuido a este periférico, desde que não use o comando CLOSE # *3.

TOP 10 EM INGLATERRA

(Tirado da revista «YOUR SPECTRUM», Abril 85)

1 — DALEY THOMPSON'S DECATHLON
2 — UNDERWURLDE
3 — SABREWULF
4 — MATCH DAY
5 — KNIGHT LORE
6 — JET SET WILLY
7 — TLL
8 — MONTY MOLE
9 — SKOOL DAZE
10 — GHOSTBUSTERS

NO CLUB Z80

(Os mais vendidos)

1 — CYCLONE
2 — DEUS EX MACHINA
3 — LAZY JONES
4 — OLYMPIMANIA
5 — GOHSTBUSTERS
6 — TURMOIL
7 — BEAR GEORGE
8 — TRAVEL WITH TRASHMAN
9 — KNIGHT LORE
10 — SYSTEM 15000

ATENÇÃO

NO PROGRAMA VIGAS CONTÍNUAS, ONDE ESTÁ... MM EM KGFXM2 DEVE LER-SE KGFXM.

MICROCOPI — Programa de Cópia

Sem ocupar memória

Adapt. da Revista Microhobby/LORENZO CEBEIRA

Fazer cópias de segurança dos programas é, àparte ser um costume são, uma necessidade. Para isso oferecemos-lhe Microcopi, programa em código máquina, que dá uma grande ajuda para este tipo de trabalho.
Um aspecto ainda a destacar do MICROCOPI é que não ocupa lugar na memória excepto o que utiliza na apresentação visual.
No mundo dos computadores existem algumas lições que convém aprender logo se não queremos que seja o nosso micro a encarregar-se de as ensinar. Uma das mais importantes é relativa às cópias de segurança. Confiar os nossos programas a uma só cassete ou a um só cartucho de Microdive é pecar por optimismo.
Por isso convém desde o primeiro momento a guardar em duplicado todos os programas e arquivos de dados.
Isto não supõe nenhum problema quando se trata dos nossos próprios programas, mas quando se trata de fazer cópias de segurança de programas comercializados, isto torna-se difícil. A maior parte deles estão protegidos de tal forma que é impossível pará-los e portanto copiá-los com o comando Save do Spectrum. Para solucionar este problema, oferecemos-lhe MICROCOPI, um programa em código máquina que lhe permitirá copiar uma parte dos programas comerciais.

Questão de ordem

Mas antes de prosseguirmos, temos de fazer uma advertência: fazer cópias de segurança dos programas que compramos, é uma actividade perfeitamente legal e recomendada inclusive por muitas firmas de Software.
Em contrapartida fazer cópias de pogramas que compramos, para vender ou para dar a outras pessoas, é uma actividade pouco ética e provavelmente ilegal, mas a legislação sobre propriedade intelectual deixa muito a desejar nesta matéria. Em qualquer caso fique bem claro que a finalidade deste programa é unicamente a primeira citada.

Características do programa

MICROCOPI permite copiar qualquer programa com Header (logo explicaremos isto) cujo comprimento não supere os 8973 bytes no Spectrum de 16K, ou 41741 bytes no modelo 48K. Também pode copiar um programa que conste de várias partes, se nenhuma delas superar o comprimento indicado. Para copiar este tipo de programas é necessário fazê-lo parte a parte.
Além disso, MICROCOPI proporciona o tipo, nome, comprimento e começo do programa que está a carregar ou a gravar e permite repetir estas operações quantas vezes desejarmos.

Fazer cópias de segurança dos programas que compramos é uma actividade perfeitamente legal.

Depois de carregado aparece no écran o menu com as opções «LOAD», «SAVE» e «FIN». Premindo a tecla correspondente o programa realizará a operação requerida. Observa-se que o F de «FIN» está em maiúscula e para isso deve premir simultâneamente Caps Shift e a tecla. Esta operação é uma protecção contra descuidos e tem como efeito apagar o programa da memória e regressar à posição inicial. As outras duas opções explicam-se por si mesmas. Premindo a tecla «LOAD» entrará em modo de carga. Ponhamos em marcha a fita com o programa que queremos copiar e enquanto MICROCOPI tiver copiado o Header, dar-nos-á os dados do programa.
Se esse não é o que queremos copiar, premindo Break, MICROCOPI passará a carregar o programa seguinte da fita. Uma vez carregado e se tudo correu bem, a mensagem «correcto» aparecerá no écran. Premindo Enter voltaremos ao menu onde podemos escolher a opção Save para gravar o programa. Pode terminar a operação de gravação fazendo Break ou então continuar a fazer o número de cópias que desejar do mesmo programa.

Informação

Vejamos agora o que significa a informação que aparece no écran em primeiro lugar. Temos o tipo de programa que pode ser:
«Program», «Number array», «Character array» ou «Bytes». Se tem alguma dúvida neste campo consulte o seu manual. Segue-se o nome do programa onde podem ser utilizados códigos de controle ou caracteres especiais.
Quando MICROCOPI encontra algum destes, substitui-os por um sinal de interrogação, ou de contrário o nome poderia imprimir-se na parte superior do écran provocando um resultado catastrófico.

MICROCOPI proporciona o tipo, nome, comprimento e começo do programa

Encontramo-nos com um número que indica o comprimento do programa que estamos a carregar. Se este é superior ao que indicamos anteriormente, MICROCOPI parará imediatamente e indicará que não é possível copiá-lo, com a mensagem «NO CABE».
Convém ter em conta, que em ocasiões se utilizam Headers «falsos» que indicam um comprimento diferente do real. Podemos encontrar a mensagem de «Error de carga» em programas que usam este tipo de protecção.

Outros truques

Existem muitos outros truques de protecção, como gravar os programas sem Header, colar os blocos de dados, etc. . . . e MICROCOPI não copia este tipo de programas. Note ainda que podemos ter uma cópia que aparentemente se fez sem problemas mas que não funciona.
Embora isto seja bastante raro, é perfeitamente possível que aconteça. Os truques de protecção são infinitos e não existe nenhum programa de cópia que possa com todos eles. Em contrapartida não existe nenhum programa que não possa ser copiado. E seguindo com o nosso tema vejamos agora o sinificado de «Comienzo» — Começo.
Se se trata de um programa Basic, este número indica a linha de execução automática com que foi gravado o programa. Quando não aparece significa que não existe linha de autoexecução. Se se trata de Bytes será indicada a posição do código máquina na memória.
Por último no caso de «Array», este número não aparece.

Como se faz

Passemos agora à parte prática do tema. Em primeiro lugar temos que introduzir o programa número 1 e gravá-lo em cassete com a ordem:

SAVE «MICROCOPI» Line 10. A parte mais difícil do trabalho vem a seguir. Temos que introduzir o programa listado número 2 procurando não cometer erros.

Convém ter cuidado especial com o último número de cada linha DATA.

Trata-se do controle que nos permitirá corrigir os erros que tenhamos cometido (recorde a lei de Murphy). Se tem alguma dúvida talvez o ajude a saber que nestas linhas só se aplicam números e letras maiúsculas de A a F.

Uma vez terminada a listagem convém fazer uma cópia de segurança do mesmo numa cassete àparte.

A seguir podemos pôr o programa em funcionamento com o comando RUN.

Se tudo correu bem, ao fim de 1 minuto e alguns segundos, aparecerá a conhecida mensagem «Start tape, then press any key», indicando-nos que podemos gravar o código máquina que o programa produziu.

Naturalmente gravá-lo-emos a seguir ao programa da listagem 1. Se cometermos algum erro nas linhas DATA, o programa parará indicando-nos o número da linha com erro, para que efectuemos a correcção correspondente.

Erros

Agora tecle LOAD «MICROCOPI» ou «LOAD» para carregar o programa. Há uma possibilidade mínima do programa não funcionar. Se tivermos cometido erros que se anulem entre si, o programa não poderá detectá-los. Neste caso não haverá outra solução que não seja a de voltar a verificar a listagem. Uma vez tendo tudo em ordem deve fazer 2 cópias do programa MICROCOPI.

I Programa

```
  10 BORDER 0: INK 0: PAPER 0: F
LASH 0: BRIGHT 0: OVER 0: INVERS
E 0
  20 LET TOP=PEEK 23732+256*PEEK
 23733
  30 CLEAR TOP
  40 LOAD "copy"CODE 16384
  50 RANDOMIZE USR 17010
```

II Programa

```
   1 REM ****COPIADOR****
  10 DATA "00000000000000000000"
,0
  20 DATA "0000000000000000160C"
,34
  30 DATA "01110110071200 4C4F4E"
,293
  40 DATA "474954554 43A20FF160E"
,762
  50 DATA "011101100712 00434F4D"
,283
  60 DATA "49454E5A4F3A20FF1612"
,774
  70 DATA "01110210071200206C3D"
,262
  80 DATA "6C6F6164202020202073"
,691
  90 DATA "3D736176652020202020"
,652
 100 DATA "463D66696E2020FF1612"
,807
 110 DATA "01110210071201434152"
,276
 120 DATA "47414E444F1200202042"
,509
 130 DATA "7265616B207061726120"
,903
 140 DATA "7061726172FF16120111"
,847
 150 DATA "02100712014752414241"
,393
 160 DATA "4E444F12002020427265"
,588
 170 DATA "616B2070617261207061"
,897
 180 DATA "726172FF161201110010"
,654
 190 DATA "00120020202020202020"
,242
 200 DATA "20202020202020202020"
,320
 210 DATA "20202020202020202020"
,320
 220 DATA "20202020FF1612011102"
,443
 230 DATA "10071201455252 4F5220"
,468
 240 DATA "44452043415247411200"
,537
 250 DATA "202070756C736520456E"
,828
 260 DATA "746572FF161201110210"
,662
 270 DATA "071201434F5252454354"
,555
 280 DATA "4F1200202070756C7365"
,714
 290 DATA "20456E746572FF161201"
,838
 300 DATA "110210071201 4E4F2043"
,317
 310 DATA "4142451200202070756C"
,619
 320 DATA "736520456E746572FF06"
,1019
 330 DATA "0FCD440EC9626B010000"
,709
 340 DATA "03233EFFBE20F9CD3C20"
,1123
 350 DATA "C9FB213B5CCB6E28F8C9"
,1438
 360 DATA "FBAF213B5CCB6E28F73A"
,1268
 370 DATA "085CCBAEFE0D20EEC900"
,1215
 380 DATA "00000000000000000000"
,0
 390 DATA "00000000000000000000"
,0
 400 DATA "00000000000000000000"
,0
 410 DATA "0000000000000000 7F00"
,127
 420 DATA "00000000000000000000"
,0
 430 DATA "00000000000000000000"
,0
 440 DATA "00160000000000000000"
,27
 450 DATA "00000000000000000000"
,0
 460 DATA "00000000000000000000"
,0
 470 DATA "000000000000 7F7F7F7F"
,508
 480 DATA "7F007F007F7F007F7F00"
,762
 490 DATA "7F7F7F002424002D2D2D"
,588
 500 DATA "003636360001B00007F00"
,318
 510 DATA "7F007F007F007F00007F"
,635
 520 DATA "00007F007F002400002D"
,335
 530 DATA "002D003600360001B0000"
,180
 540 DATA "7F0000007F007F007F7F"
,635
 550 DATA "007F00007F7F7F002424"
,580
 560 DATA "002D2D2D0036363600 1B"
,324
 570 DATA "00000000000000000000"
,0
 580 DATA "00000000000000000000"
,0
```

```
590 DATA "00000000000000360000"
,54
600 DATA "00000000000000000000"
,0
610 DATA "00000000000000000000"
,0
620 DATA "00000000000000000000"
,0
630 DATA "0000000001AF216A5C"
,407
640 DATA "772AB25C2B2B36422B36"
,734
650 DATA "9B3E02CD0116AF321140"
,753
660 DATA "1171412100581AFE01CA"
,799
670 DATA "9E427723131BF5CD3F41"
,999
680 DATA "113A40CD4541FBAF213B"
,995
690 DATA "5CCB6E28F73A085CCBAE"
,1227
700 DATA "FE46CC0000FE6CCA0343"
,1162
710 DATA "FE7320E43A1140FE0028"
,1062
720 DATA "DD11AE40CD4541118840"
,1032
730 DATA "CD4541DD210040111100"
,691
740 DATA "AFCDC2040632FB7610FD"
,1272
750 DATA "3EFFED5B0B40DD2A535C"
,1158
760 DATA "CDC20411AE40CD454111"
,1014
770 DATA "FE40CD4541CD5541C39E"
,1365
780 DATA "42CD3F41116240CD4541"
,917
790 DATA "AF321140111100DD2100"
,594
800 DATA "4037CD560530E63A0040"
,815
810 DATA "FE0430DF3E16D73E09D7"
,1114
820 DATA "3E01D73A004011C009CD"
,823
830 DATA "0A0CC33C433E3FC9060A"
,685
840 DATA "110040131AFE20DC3943"
,755
850 DATA "FE80D43943D710F11112"
,1225
860 DATA "40CD4541ED4B0B40CD2B"
,1038
870 DATA "2DCDE32D3A0040FE03CA"
,1103
880 DATA "7A43FE00C28A43ED4B0D"
,1167
890 DATA "40210F27A7ED42DA8A43"
,1044
900 DATA "112640CD4541ED4B0D40"
,847
910 DATA "CD2B2DCDE32D2AB25C01"
,1083
920 DATA "2800A7ED42ED4B535CA7"
,1164
930 DATA "ED42ED4B0B40A7ED42D2"
,1370
940 DATA "B64311AE40CD4541111F"
,891
950 DATA "41CD4541CD5E41C39B42"
,1184
960 DATA "3EFFDD2A535CED5B0B40"
,1158
970 DATA "37CD5605DAD94311AE40"
,1108
980 DATA "CD454111D740CD4541CD"
,1179
990 DATA "5E41C39B4211AE40CD45"
,1104
1000 DATA "4111FE40CD45413E0132"
,852
1010 DATA "1140CD5E41C39E420000"
,864
9000 CLEAR 30999: RESTORE
9010 LET A=10: LET B=11: LET C=1
2: LET D=13: LET E=14: LET F=15
9020 LET direccao=31000
9030 FOR n=10 TO 1010 STEP 10
9040 READ h$
9050 LET contador=0
9060 FOR m=1 TO 19 STEP 2
9070 LET byte=16*VAL h$(m)+VAL h
$(m+1)
9080 LET contador=contador+byte
9090 POKE direccao,byte
9100 LET direccao=direccao+1
9110 NEXT m
9120 READ control
9130 IF contador<>control THEN P
RINT "Erro na linha ";n: STOP
9140 NEXT n
9150 PRINT "Correcto.Retire o ca
bo Ear e     grave o codigo maqui
na"
9160 SAVE "copy"CODE 31000,1010
9170 PRINT "Por de novo o cabo E
ar para      verificar a gravacao
"
9180 VERIFY "copy"CODE 31000,101
0
9190 PRINT "A gravacao esta corr
ecta"
9200 STOP
```

FAÇA O SEU PRÓPRIO COPIADOR

CARLOS MORENO
PORTO

(Continuação do número anterior)

Como sabem depois de transferir uma rotina C/M de um local para outro, todas as instruçoes JP têm de ser alteradas. Por isso faça poke 65495,255 para corrigir o endereço de uma JP da rotina de load que foi transferida.

Feito isto a rotina está pronta a ser utilizada, no entanto antes de a executar faça SCF para indicar a rotina que vai fazer load e não verify (ser a flag carry for zero a rotina pensará que quer fazer verify, o que pode ser utilizado para verificar se os bytes a copiar entraram correctamente para a memória).

Vejamos agora a listagem de um exemplo da utilização da rotina para copiar programas (bloco a bloco):

```
ld ix, 16384     ; endereço para onde vamos apanhar os bytes
ld de, 48976     ; quantidade de bytes que podemos apanhar
                   sem corromper a rotina de load
scf              ; sinalizar que vamos fazer load
call 65366       ; chamar a rotina de load
ld (65534), ix   ; ix contém o endereço para onde foi apanhado
                   o último byte
ld hl, (65534)   ; dar a hl o valor de ix
ld bc, 16384     ; dar a bc o valor do endereço inicial
and a            ; tornar a flag carry zero para a subtracção que
                   se segue
sbc hl, bc       ; agora hl tem o número de bytes apanhados
ex hl, de        ; Dar esse valor a de
call + 1f54      ; chamar a rotina de verificação de break
jr c, 253        ; esperar até break e caps shift serem premidas
                   simultâneamente
ld a, (16384)    ; dar a a o valor do leading byte
```

```
ld ix, 16385 ; dar a ix o valor do endereço da base do bloco a
               copiar, este valor é 16385 e não 16384 porque
               este último endereco tem o valor do leading
               byte
call +04C2   ; chamar a rotina de save
jp 65507     ; voltar ao início
```

Esta rotina deve ser assemblada a partir do endereço 65507, se por alguma razão modificar esse endereço tenha cuidado com a instrução jp do fim da rotina

Esta listagem é apenas o exemplo da utilização da rotina de load, podendo ser substituida por outra que esteja mais de acordo com as suas necessidades.

Agora tudo depende da sua imaginação.

FIM

TESTE VOCACIONAL

(Continuação do número anterior)

```
1050 DATA "Qual das seguintes ac
ções gosta-rias mais de fazer no
 dia do teuaniversário:","Ir com
 os teus amigos visitar  um mus
eu de botanica,um gran-  de aqu
ário ou um planetário","Organiza
r um concurso entre      os teus
convidados","Relatar aos teus co
nvidados a   historia da tua vid
a,utili-     zando diapositivos,
quadros      humoristicos ou uma
 redacção     bem humorada"
1060 DATA "Se te dessem um prémi
o gostariasque fosse:","Por tere
s escrito um livro      sobre a
vida humana","Por teres inventad
o uma          máquina","Por tere
s feito uma obra de      arte"
1070 DATA "Quando visitas uma po
voação que não conheces que pref
eres fazer?","Observar as casas,
edificios        ou monumentos","Ob
servar a forma de viver das    pe
ssoas,os seus hábitos e        co
mportamentos","Travar conhecimen
to com pes-      soas da povoacao
e interrogá-     -las sobre coisas
 que te          interessam"
1080 DATA "Imagina  que  te  enc
ontras  em  frente  de  um  grand
e rio; que preferias fazer?","Ca
nalizá-lo para uma albufei-    ra
 e depois construir uma        ba
rragem para produzir ener-     gi
a eléctrica","Canalizá-lo para z
onas agri-     colas e montar sis
temas de        regadio","Fazer um
 belo quadro do rio e   da paisa
gem à volta"
1090 DATA "Supõe que te encontra
s em frentede um monte de cacos
velhos:    Brinquedos,roupas móv
eis e ou-  tros objectos.Que te
ocorreria  fazer,de preferência:
","Recolher tudo e limpar aquele
   local tornando-o bonito","Pro
curar peças que te permi-   tis
sem construir um novo        bri
nquedo","Aproveitar aquelas cois
as que   te pudessem servir para
 fazer  objectos de decoração"
1100 DATA "Dos seguintes element
os escolhe aquele que tiver mais
 significa-do para ti:","Uma esc
ultura","Uma máquina","Uma plant
a"
1110 DATA "Dão-te  a  escolher
entre  os  seguintes museus,para
 vistar:","Museu de Aeronáutica"
,"Museu da Natureza","Museu do t
eatro"
1120 DATA "Escolhe entre:","Inve
stigar","Organizar","Comunicar"
1130 DATA "Que preferirias fazer
:","Uma grande escultura","Um ja
rdim zoológico","Uma ponte de fe
rro sobre um         rio"
1140 DATA "Preferes dar um passe
io:","Sozinho","Com dois ou três
 amigos","Com muitas pessoas"
1150 DATA "Imagina uma das segui
ntes metas para as tuas activida
des.         Qual escolherias de p
referência:","Saúde","Beleza","R
iqueza"
1160 DATA "Escolhe entre:","Info
rmar pessoas","Cuidar das pessoa
s","Organizar o trabalho das
    pessoas"
1170 DATA "Escolhe entre:","Fala
r em público","Construir um bone
co de barro","Cuidar de uma cria
nça doente"
1180 DATA "Aonde preferes ir:","
A um jardim com muitas plan-
tas,lagos e pessoas","A uma fábr
ica com robôs","A um bairro pobr
e"
1190 DATA "Escolhe entre:","Faze
r","Contemplar","Falar"
1200 DATA "Escolhe entre:","A te
rra","O ar","O fogo"
1210 DATA "Escolhe entre:","Cria
r animais","Pintar animais","Faz
er parte de uma sociedade    pro
tectora de animais"
1220 DATA "Preferirias:","Fazer
medicamentos","Fazer máquinas pa
ra produzir    medicamentos","Pa
rticipar na organização da    as
sistencia médica"
1230 DATA "O que é que mais admi
ras:","Um grupo de deficientes q
ue      se organizam para se defe
nder    melhor na sociedade","Um
grupo de bailado","Um grupo de e
cologistas"
1240 DATA "Que gostarias mais de
 aprender  fazer:","Um belo edif
ício","Uma grande empresa de
       publiciade","Um conjunto
de máquinas"
1250 DATA "A que dás mais valor:
","Ao entendimento entre as
   pessoas","Ao desenvolvimento
industrial","A conservação da Na
tureza"
1260 DATA "Escolhe entre:","Faze
r uma lei","Fazer um discurso","
Fazer um quadro"
2000 DATA r$(5)="a",r$(6)="a",r$
(8)="b",r$(10)="c",r$(11)="b",r$
(12)="a",r$(13)="b",r$(14)="a",r
$(15)="a",r$(16)="b",r$(17)="c",
r$(18)="a",r$(19)="b",r$(20)="a"
,r$(21)="a",r$(22)="a",r$(23)="c
",r$(25)="c"
2010 DATA r$(2)="b",r$(3)="b",r$
(4)="c",r$(6)="b",r$(7)="a",r$(8
)="a",r$(9)="b",r$(10)="b",r$(11
)="a",r$(12)="a",r$(13)="c",r$(1
4)="a",r$(18)="b",r$(19)="a",r$(
20)="b",r$(22)="b",r$(24)="c",r$
(25)="b"
2020 DATA r$(1)="b",r$(3)="c",r$
(5)="b",r$(7)="b",r$(7)="b",r$(9
)="a",r$(12)="b",r$(14)="c",r$(1
5)="c",r$(16)="c",r$(17)="a",r$(
18)="c",r$(19)="a",r$(21)="c",r$
(22)="c",r$(23)="a",r$(24)="b",r
$(25)="a",r$(26)="a"
2030 DATA r$(1)="c",r$(2)="a",r$
(3)="c",r$(4)="a",r$(5)="c",r$(6
)="a",r$(7)="c",r$(11)="c",r$(12
)="c",r$(14)="b",r$(16)="a",r$(1
7)="a",r$(18)="c",r$(19)="c",r$(
20)="c",r$(24)="b",r$(25)="a",r$
```

```
(26)="b"
2040 DATA r$(1)="a",r$(2)="c",r$
(3)="a",r$(4)="b",r$(6)="c",r$(7
)="a",r$(8)="c",r$(9)="c",r$(10)
="a",r$(13)="a",r$(15)="b",r$(17
)="b",r$(19)="a",r$(20)="c",r$(2
1)="b",r$(23)="b",r$(24)="a",r$(
26)="c"
```

FIM

FICHEIRO DE CASSETTES

Autor: MANUEL JOSÉ QUINAZ
PORTO

INSTRUCOES:

Este programa tem a capacidade de guardar na memoria 150 fichas de programas com os seguintes campos:
NOME,Memoria,Numero da cassete em que se encontra,Lado da cassete e Observacoes (ate 128 caracteres).

Para introduzir o program proceda do seguinte modo:
Meta a Listagem 1 do programa em Basic gravando-a de seguida com SAVE "Ficheiro" LINE 10 (ATENCAO:Nao experimente o programa sem o codigo maquina metido)
Em seguida faca PRINT USR 0 e de entrada do Loader da Listagem 2
Depois do programa metido faca RUN e de entrada do endereco onde quer comecar a meter o codigo maquina.De entrada dos varios valores,e quando quiser terminar entre com 999.

O programa tem varios tipos de caracteres e isso ocupa muitos bytes e demora muito tempo a passar para o computador para evitar isso nao passe o codigo maquina correspondente as partes Letras 1,Letras 2,Letras 3,Letras 4,escrever nas Linhas 9920,9930,9940,9950,9960 apenas o comando RETURNE.

Se nao quiser passar o programa envie um pedido por carta e a quantia de 160$00 em vale ou cheque ou em selos de 20$00 e sera remetida uma copia do programa.

FICHEIRO DE PROGRAMAS

1 - DAR ENTRADA DE FICHAS
2 - CONSULTAR O FICHEIRO
3 - APAGAR UMA FICHA
4 - DADOS PARA A PRINTER
5 - GRAVAR FICHEIRO
6 - VERIFICAR TODO FICHEIRO

A sua opcao->

© Copyright,1984-Programa elaborado por Manuel Jose Quinaz

FICHAS PREENCHIDAS-0

Programa Numero:1 10021

Jetpac

Zx Spectrum 16 K

Cassete No.02 Lado 1

OBSERVACOES

->Trata-se de um jogo de accao em que voce tera de guiar o astronauta ajudando-o a construir o Foguetao.

```
  10 CLEAR 59999: LOAD ""CODE
  50 DIM o$(150,128): DIM n$(150
,30): DIM m$(150,4): LET f=1
  60 BORDER 1: PAPER 1: INK 7: C
LS
  70 LET d$="FICHEIRO DE PROGRAM
AS": LET y=0: LET x=40: LET l=1:
 LET a=2: GO SUB 9940: GO SUB 99
00: GO SUB 9960
  75 PLOT 35,159: DRAW 175,0
  80 GO SUB 9950: INK 4: PRINT A
T 4,3;"1- dar entrada de fichas"
''TAB 3;"2- consultar o ficheiro
"''TAB 3;"3- apagar uma ficha"''
TAB 3;"4- dados para a printer"'
'TAB 3;"5- gravar ficheiro"''TAB
 3;"6- verificar todo ficheiro";
#0;TAB 3;"fichas preenchidas-";
f-1: GO SUB 9960: INK 7
  90 LET d$="© Copyright,1984-Pr
ograma elaborado por Manuel Jose
 Quinaz": INK 5: LET x=2: LET y=
21: GO SUB 9910: INK 7
  91 PLOT 0,50: DRAW 255,0: DRAW
 0,125: DRAW -255,0: DRAW 0,-125
: INK 5: PLOT 0,10: DRAW 255,0:
DRAW 0,-10: DRAW -255,0: DRAW 0,
10: INK 7: LET d$="A sua opcao->
": LET x=20: LET y=135: LET l=2:
 LET a=3: GO SUB 9920: GO SUB 99
00: GO SUB 9960
  95 LET d$=INKEY$: IF CODE d$<4
9 OR CODE d$>54 THEN GO TO 95
  96 GO SUB 9920: LET x=230: GO
SUB 9900: GO SUB 9960
  98 LET x=30: LET y=5: LET l=10
: GO SUB 9970
 100 CLS : GO SUB (CODE d$-48)*1
000
 200 GO TO 60
1000 REM Entrada de fichas
1001 INK 1: PAPER 5: BORDER 5: C
LS : IF f=151 THEN PRINT FLASH 1
;AT 10,7;"FICHEIRO COMPLETO": LE
T x=70: LET y=3: LET l=20: GO SU
B 9970: RETURN
1010 LET d$="Programa Numero:"+S
TR$ f: LET x=0: LET a=2: LET l=1
: LET y=10: GO SUB 9900
1015 PLOT 0,145: DRAW 255,0
1020 INPUT "Nome do Programa:";q
$: IF LEN q$>30 THEN GO TO 1020
1030 LET n$(f)=q$: LET d$=q$: LE
T x=0: LET y=37: LET a=3: LET l=
1: GO SUB 9900
1040 INPUT "Memoria:";m: IF m<>1
6 AND m<>48 THEN GO TO 1040
1041 LET d$="Zx Spectrum "+STR$
m+" K": LET x=0: LET y=70: LET l
=1: LET a=2: GO SUB 9900: LET v$
=STR$ (m=48)
1045 INPUT "Cassete No.";t: LET
t$=STR$ t
1047 INPUT "Lado:";s: IF s<>1 AN
D s<>2 THEN GO TO 1047
1048 LET s$=STR$ s
1049 IF LEN t$<2 THEN LET t$="0"
+t$
1050 LET m$(f)=v$+t$+s$
1060 LET d$=STR$ f+m$(f): LET x=
198: LET y=10: LET l=1: LET a=2:
 GO SUB 9900
1070 LET d$="Cassete No."+t$: LE
```

NOTA: Os blocos em código máquina correspondentes aos vários tipos de caracteres (letras 1,2,3,4) serão publicadas na próxima revista.

```
T l=1: LET a=2: LET x=0: LET y=9
0: GO SUB 9900: LET d$="Lado "+s
$: LET x=207: GO SUB 9900: PLOT
0,69: DRAW 255,0
1080 INPUT "Observacoes:";c$: IF
 LEN c$>128 THEN GO TO 1080
1085 LET d$="observacoes": GO SU
B 9950: LET x=40: LET a=2: LET l
=2: LET y=115: GO SUB 9900: GO S
UB 9960
1090 LET o$(f)=c$: LET d$="->"+c
$: LET x=0: LET y=18: INK 1: GO
SUB 9910: INK 0
1100 INPUT "Alguma alteracao (s/
n)";d$
1110 IF d$="s" THEN GO TO 1000
1120 IF d$="n" THEN GO TO 1130
1125 GO TO 1100
1130 LET f=f+1
1135 INPUT "Mais alguma ficha?(s
/n)";d$: IF d$="s" THEN GO TO 10
00
1140 IF d$="n" THEN RETURN
1150 GO TO 1135
2000 REM Consultar ficheiro
2001 INK 0: PAPER 4: BORDER 4: C
LS
2005 IF f=1 THEN PRINT AT 10,5;
FLASH 1;"NAO EXISTE NENHUMA FICH
A": LET x=40: LET y=4: LET l=40:
 GO SUB 9970: RETURN
2010 LET d$="Consultar o ficheir
o": LET l=1: LET a=2: LET y=0: L
ET x=40: GO SUB 9930: GO SUB 990
0: GO SUB 9960
2020 GO SUB 9950: PRINT AT 4,3;"
1-procurar pelo nome"''TAB 3;"2
- procurar pelo numero": GO SUB
9960
2030 LET d$="A sua opcao:": LET
l=2: LET a=1: LET x=24: LET y=75
: GO SUB 9920: GO SUB 9900
2040 LET d$=INKEY$: IF d$<>"1" A
ND d$<>"2" THEN GO TO 2040
2050 LET x=216: GO SUB 9900: GO
SUB 9960
2060 IF d$="2" THEN GO TO 2200
2070 INPUT "Nome do programa:";d
$: LET d=LEN d$
2080 FOR c=1 TO f-1: IF n$(c)( T
O d)=d$ THEN GO TO 2500
2090 NEXT c: CLS : PRINT AT 10,1
0; FLASH 1;"INEXISTENTE": LET x=
30: LET y=2: LET l=50: GO SUB 99
70: GO TO 2900
2200 INPUT "Numero do programa "
;c: IF c>=f THEN CLS : PRINT AT
10,10; FLASH 1;"INEXISTENTE": LE
T x=30: LET y=2: LET l=50: GO SU
B 9970: GO TO 2900
2500 INK 0: PAPER 5: BORDER 5: CLS
2510 LET d$=STR$ c+m$(c): INK 2:
 LET x=198: LET y=10: LET l=1: L
ET a=2: GO SUB 9900
2520 LET d$="Programa Numero:"+S
TR$ c: LET x=0: LET a=2: LET l=1
: LET y=10: GO SUB 9900: INK 0
2530 LET d$=n$(c): LET x=0: LET
y=37: LET a=3: LET l=1: GO SUB 9
900
2540 IF m$(c,1)="0" THEN LET i$=
"16 K": GO TO 2550
2541 LET i$="48 K"
2550 LET d$="Zx Spectrum "+i$: L
ET x=0: LET y=70: LET l=1: LET a
=2: GO SUB 9900
2570 LET d$="Cassete No."+m$(c)(
2 TO 3): LET l=1: LET a=2: LET x
=0: LET y=90: GO SUB 9900: LET d
$="Lado "+m$(c,4): LET x=207: GO
 SUB 9900: PLOT 0,69: DRAW 255,0
2580 INK 1: LET d$="observacoes"
: GO SUB 9950: LET x=40: LET a=2
: LET l=2: LET y=115: GO SUB 990
0: GO SUB 9960: INK 0
2590 LET c$=o$(c): LET d$="->"+c
$: LET x=0: LET y=18: INK 1: GO
SUB 9910: INK 0
2600 PLOT 0,145: DRAW 255,0: PLO
T 34,42: DRAW INK 1;180,0
2610 INPUT "Quer copiar (s/n)";d
$: IF d$="s" THEN COPY
2900 INPUT "Mais alguma procura
(s/n)";d$: IF d$="n" THEN RETURN

2910 IF d$="s" THEN GO TO 2000
2920 GO TO 2900
3000 REM Apagar uma ficha
3010 INK 2: PAPER 6: BORDER 6: C
LS
3020 LET d$="Apagar uma ficha":
GO SUB 9930: LET x=50: LET y=0:
LET l=1: LET a=2: GO SUB 9900: G
O SUB 9960
3030 INPUT "Ficha a apagar ";c:
IF c>=f THEN PRINT AT 10,7; FLAS
H 1;"FICHA NAO PREENCHIDA": LET
x=50: LET y=3: LET l=30: GO SUB
9970: RETURN
3040 LET d$=n$(c): LET x=0: LET
y=30: LET l=1: LET a=2: GO SUB 9
900
3050 INPUT "Tem a certeza (s/n)"
;d$: IF d$="n" THEN RETURN
3060 IF d$<>"s" THEN GO TO 3050
3070 LET o$(f)="": LET m$(f)="":
 LET n$(f)="": FOR i=c TO f-1: L
ET n$(i)=n$(i+1): LET o$(i)=o$(i
+1): LET m$(i)=m$(i+1): NEXT i:
LET f=f-1
3080 LET d$="APAGADA": LET l=2:
LET a=4: LET y=120: LET x=50: GO
 SUB 9900: LET x=50: LET y=5: LE
T l=20: GO SUB 9970: RETURN
4000 REM Dados para a printer
4010 INK 4: PAPER 1: BORDER 1: C
LS
4020 LET d$="dados para a printe
r": LET x=40: LET y=0: LET l=1:
LET a=2: GO SUB 9930: GO SUB 990
0: GO SUB 9960
4030 GO SUB 9950: PRINT AT 4,3;"
1- saida de observacoes"''TAB 3;
"2- saida de programas": GO SUB
9960
4040 LET d$="A sua opcao:": LET
x=23: LET y=70: LET l=2: LET a=1
: GO SUB 9920: GO SUB 9900
4050 LET d$=INKEY$: IF d$<>"1" A
ND d$<>"2" THEN GO TO 4050
4060 LET x=213: GO SUB 9900: GO
SUB 9960
4070 LET x=40: LET y=4: LET l=10
: GO SUB 9970
4080 CLS : IF d$="2" THEN GO TO
4500
4090 INPUT "Em 64 ou 32 colunas
?";d
4100 IF d<>64 AND d<>32 THEN GO
TO 4090
4110 IF d=64 THEN GO TO 4300
4120 INPUT "Todos os programas (
s/n)";d$: IF d$<>"s" AND d$<>"n"
 THEN GO TO 4120
4130 IF d$="s" THEN GO TO 4200
4140 INPUT "Numero do programa "
;d: IF d>=f THEN PRINT AT 10,10;
 FLASH 1;"INEXISTENTE": LET x=40
: LET y=2: LET l=20: GO SUB 9970
: RETURN
4150 LPRINT o$(d): RETURN
4200 FOR d=1 TO f-1: LPRINT o$(d
): NEXT d: RETURN
4300 INPUT "Todos os programas (
s/n)";d$: IF d$<>"s" AND d$<>"n"
 THEN GO TO 4300
4310 IF d$="s" THEN GO TO 4400
4320 INPUT "Numero do programa "
;d: IF d>=f THEN PRINT AT 10,10;
 FLASH 1;"INEXISTENTE": LET x=40
: LET y=2: LET l=20: GO SUB 9970
: RETURN
4330 LET d$=n$(d): LET x=0: LET
y=0: GO SUB 9910: LET d$=o$(d):
LET x=0: LET y=2: GO SUB 9910: C
OPY : RETURN
4400 LET x=0: LET y=0: FOR i=1 T
O f-1: LET d$=n$(i): GO SUB 9910
```

```
: LET y=y+2: LET d$=o$(i): GO SU
B 9910: LET y=y+3
4410 IF i/4=INT (i/4) THEN LET y
=0: COPY : CLS
4420 NEXT i
4430 COPY : RETURN
4500 FOR i=1 TO f-1 STEP 2: LPRI
NT n$(i)( TO 16);n$(i+1)( TO 16)
: NEXT i
4510 RETURN
5000 REM Gravar ficheiro
5010 INK 3: PAPER 6: BORDER 6: C
LS
5020 LET d$="Gravar o ficheiro":
 LET x=40: LET y=0: GO SUB 9930:
 LET l=1: LET a=2: GO SUB 9900:
GO SUB 9960
5030 GO SUB 9950: PRINT AT 4,3;"
1- gravar em diskete"'TAB 3;"2-
 gravar em cassete": GO SUB 9920
: LET d$="A sua opcao:": LET l=2
: LET a=1: LET y=70: LET x=20: G
O SUB 9900
5040 LET d$=INKEY$: IF d$<>"1" A
ND d$<>"2" THEN GO TO 5040
5050 LET x=213: GO SUB 9900: GO
SUB 9960
5060 LET x=30: LET y=2: LET l=30
: GO SUB 9970: CLS
5070 IF d$="2" THEN GO TO 5500
5080 LET d$="Prepare a diskete
 onde quer": LET x=0: LET y=0:
 LET l=1: LET a=2: GO SUB 9900:
LET y=16: LET d$="gravar o fiche
iro e diga o nome": GO SUB 9900
: LET y=32: LET d$="do ficheiro.
": GO SUB 9900
5090 INPUT "Ficheiro "; LINE k$:
 IF LEN k$>10 THEN GO TO 5090
5100 LET d$="Gravando "+k$: LET
y=70: GO SUB 9900: PRINT AT 19,0
;"ATENCAO:Quando quiser carregar
 o          ficheiro gravado tera
de          fazer "CLEAR 59999"":
SAVE *"m";1;k$ LINE 9999: SAVE *
"m";1;"FicheiroCODE "CODE 60000,
5535
5110 CLS : GO SUB 9920: LET d$="
OK!": LET x=20: LET y=10: LET a=
20: LET l=10: GO SUB 9900: GO SU
B 9960
5120 LET x=60: LET y=5: LET l=30
: GO SUB 9970
5130 RETURN
5500 LET d$="Prepare a cassete
 onde quer": LET x=0: LET y=0:
 LET l=1: LET a=2: GO SUB 9900:
LET y=16: LET d$="gravar o fiche
iro e diga o nome": GO SUB 9900
: LET y=32: LET d$="do ficheiro.
": GO SUB 9900
5510 INPUT "Ficheiro "; LINE k$:
 IF LEN k$>10 THEN GO TO 5090
5520 PRINT AT 10,0;"Ponha o grav
ador a gravar e    prima uma te
cla.": PRINT AT 19,0;"ATENCAO:Qu
ando quiser carregar o        fi
cheiro gravado tera de        fa
zer "CLEAR 59999".": SAVE k$ LIN
E 9998: POKE 23794,181: SAVE "FC
CODE "CODE 60000,5535
5530 PRINT #0;AT 0,0;"Quer verif
icar (s/n)": IF INKEY$="" THEN G
O TO 5530
5535 IF INKEY$="s" THEN VERIFY "
": VERIFY ""CODE
5540 CLS : LET d$="OK!": LET x=2
0: LET y=10: LET a=20: LET l=10:
 GO SUB 9900:
5550 LET x=60: LET y=5: LET l=30
: GO SUB 9970
5560 RETURN
6000 REM Verificar o ficheiro
6010 INK 1: PAPER 5: BORDER 5: C
LS
6020 LET d$="verificar o ficheir
o todo": LET x=20: LET y=0: LET
l=1: LET a=2: GO SUB 9930: GO SU
B 9900: GO SUB 9960
6030 PRINT AT 3,0;: FOR i=1 TO f
-1: PRINT i;TAB 3;"-";n$(i)( TO
23);"-";m$(i): NEXT i
6040 PRINT #0; FLASH 1;"
PRIMA UMA TECLA
6050 IF INKEY$<>"" THEN RETURN
6060 GO TO 6050
9899 REM SUBROTINAS
9900 LET i=23306: POKE i,x: POKE
 i+1,y: POKE i+2,l: POKE i+3,a:
POKE i+4,8: LET i=i+4: LET w=LEN
 d$: FOR n=1 TO w: POKE i+n,CODE
 d$(n): NEXT n: POKE i+w+1,255:
LET w=USR 65220: RETURN : REM LE
TRAS GRANDES
9910 FOR n=1 TO LEN d$: POKE 645
99+n,CODE d$(n): NEXT n: POKE 64
599+n,0: POKE 64998,x: POKE 6499
9,y: RANDOMIZE USR 65000: RETURN
 : REM Letras Grandes
9915 REM Varios tipos de letras
9920 POKE 23606,240: POKE 23607,
234: RETURN
9930 POKE 23606,96: POKE 23607,2
38: RETURN
9940 POKE 23606,208: POKE 23607,
241: RETURN
9950 POKE 23606,64: POKE 23607,2
45: RETURN
9960 POKE 23606,0: POKE 23607,60
: RETURN
9970 POKE 60006,x: POKE 60020,y:
 POKE 60029,l: RANDOMIZE USR 600
00: RETURN
9998 LOAD ""CODE : GO TO 60
9999 LOAD *"m";1;k$CODE : GO TO
60
```

Listagem 2-M.C. LOADER

```
  10 CLEAR 59999
  20 INPUT "Primeiro endereco ";
s
  30 FOR i=s TO 65535: PRINT i;"
 ";: LET c=0: FOR n=0 TO 3: IN
PUT "valor ";a: IF a=999 THEN ST
OP
  40 LET c=c+a: PRINT a;" ";: P
OKE i+n,a: NEXT n: PRINT "=";c:
NEXT i
```

64 Colunas

```
64128   4    4    4    4    =16
64132   4    0    4    0    =8
64136   160  170  14   10   =354
64140   14   10   0    0    =24
64144   2    74   226  196  =498
64148   104  234  72   0    =410
64152   66   162  64   192  =484
64156   160  64   32   0    =256
64160   36   66   66   66   =234
64164   66   66   36   0    =168
64168   0    68   228  78   =374
64172   228  68   0    0    =296
64176   0    0    0    14   =14
64180   0    32   32   64   =128
64184   2    2    4    4    =12
64188   4    8    72   0    =84
64192   68   172  164  164  =568
64196   164  164  68   0    =396
64200   204  162  34   76   =476
64204   130  130  108  0    =368
64208   142  136  136  172  =586
64212   226  34   44   0    =304
64216   110  130  130  196  =566
64220   164  168  72   0    =404
64224   68   170  170  70   =478
64228   162  162  76   0    =400
64232   0    0    68   0    =68
64236   68   8    0    0    =76
64240   0    32   78   128  =238
64244   78   32   0    0    =110
64248   4    138  66   36   =244
64252   68   128  4    0    =200
64256   4    74   234  238  =550
64260   234  138  106  0    =478
64264   198  168  168  200  =734
```

```
64268   168 168 198 0   =534
64272   206 168 168 174 =716
64276   168 168 206 0   =542
64280   230 136 136 232 =734
64284   138 138 132 0   =408
64288   174 164 164 228 =730
64292   164 164 174 0   =502
64296   234 42  42  44  =362
64300   42  170 74  0   =286
64304   138 142 138 138 =556
64308   138 138 234 0   =510
64312   164 234 234 170 =802
64316   170 170 164 0   =504
64320   196 170 170 202 =738
64324   138 138 132 2   =410
64328   198 168 168 196 =730
64332   162 162 172 0   =496
64336   234 74  74  74  =456
64340   74  74  68  0   =216
64344   170 170 170 170 =680
64348   234 78  74  0   =386
64352   170 170 170 74  =584
64356   164 164 164 0   =492
64360   230 36  36  68  =370
64364   68  132 230 0   =430
64368   140 132 68  68  =408
64372   68  36  44  0   =148
64376   0   64  224 64  =352
64380   64  64  64  14  =206
64384   96  128 128 198 =550
64388   138 138 230 0   =506
64392   0   128 128 198 =454
64396   168 168 198 0   =534
64400   0   32  32  108 =172
64404   170 172 102 0   =444
64408   0   96  128 134 =358
64412   202 134 130 12  =478
64416   0   132 128 204 =464
64420   164 164 164 0   =492
64424   0   72  8   74  =154
64428   76  74  74  128 =352
64432   0   128 128 138 =394
64436   142 138 74  0   =354
64440   0   0   0   196 =196
64444   170 170 164 0   =504
64448   0   0   0   198 =198
64452   170 170 198 130 =668
64456   0   0   0   102 =102
64460   132 130 142 0   =404
64464   0   0   128 202 =330
64468   138 138 100 0   =376
64472   0   0   0   170 =170
64476   170 78  74  0   =322
64480   0   0   0   170 =170
64484   74  166 162 12  =414
64488   6   4   4   232 =246
64492   68  132 230 0   =430
64496   76  68  68  66  =278
64500   68  66  76  0   =212
64504   4   174 10  14  =202
64508   14  10  4   0   =28
64600   127 32  67  111 =337
64604   112 121 114 105 =452
64608   103 104 116 44  =367
64612   49  57  56  52  =214
64616   45  80  114 111 =350
64620   103 114 97  109 =423
64624   97  32  101 108 =338
64628   97  98  111 114 =420
64632   97  100 111 32  =340
64636   112 111 114 32  =369
64640   77  97  110 117 =401
64644   101 108 32  74  =315
64648   111 115 101 32  =359
64652   81  117 105 110 =413
64656   97  122 0   32  =251
64660   32  32  32  32  =128
64664   32  32  32  32  =128
64668   32  32  32  32  =128
64672   32  32  32  32  =128
64676   32  32  32  32  =128
64680   32  32  32  32  =128
64684   32  32  32  32  =128
64688   32  32  32  32  =128
64692   32  32  32  32  =128
64696   32  32  32  32  =128
64700   32  32  32  32  =128
64704   32  32  32  32  =128
64708   32  32  32  32  =128
64712   32  32  32  32  =128
64716   32  32  32  32  =128
64720   32  32  32  32  =128
64724   32  32  32  32  =128
64728   0   32  0   0   =32

64992   0   0   236 0   =236
64996   239 251 60  21  =571
65000   221 33  88  252 =594
65004   221 126 0   254 =601
65008   128 210 182 254 =774
65012   254 32  48  7   =341
65016   254 0   202 190 =646
65020   254 62  31  203 =550
65024   135 111 38  0   =284
65028   41  41  17  0   =99
65032   250 25  17  228 =520
65036   253 34  228 253 =768
65040   58  230 253 254 =795
65044   64  210 182 254 =710
65048   203 39  203 39  =484
65052   50  226 253 58  =587
65056   231 253 254 22  =760
65060   210 182 254 203 =849
65064   39  203 39  203 =484
65068   39  71  62  175 =347
65072   144 50  227 253 =674
65076   62  7   50  225 =344
65080   253 42  228 253 =776
65084   126 95  221 126 =568
65088   0   203 71  32  =306
65092   10  123 203 47  =383
65096   203 47  203 47  =500
65100   203 47  95  58  =403
65104   226 253 79  58  =616
65108   227 253 71  123 =674
65112   205 116 254 58  =633
65116   225 253 254 0   =732
65120   40  49  61  50  =200
65124   225 253 120 61  =659
65128   50  227 253 42  =572
65132   228 253 35  34  =550
65136   228 253 24  197 =702
65140   203 95  196 135 =629
65144   254 12  203 87  =556
65148   196 135 254 12  =597
65152   203 79  196 135 =613
65156   254 12  201 197 =664
65160   221 229 245 205 =900
65164   223 34  241 221 =719
65168   225 193 201 221 =840
65172   35  58  230 253 =576
65176   60  254 64  32  =410
65180   2   62  0   50  =114
65184   230 253 254 0   =737
65188   32  13  58  231 =334
65192   253 60  254 22  =589
65196   32  2   62  0   =96
65200   50  231 253 195 =729
65204   236 253 62  10  =561
65208   50  58  92  205 =405
65212   3   19  201 0   =223
65216   0   0   0   0   =0
```

Continua no próximo número

PROGRAMA BIORRITMO

PAULO METELO

Este é um programa traduzido e adaptado da ZX COMPUTING. Ele faz o seu biorritmo, bastando para isso que se diga a data em que nasceu e o mês e ano para que quer começar o seu biorritmo.

Como sabe, o biorritmo compõe-se principalmente de três ciclos, o intelectual de 33 dias, o emocional de 28 dias e ainda o físico de 23 dias, estes gráficos se o desejar podem ainda ser impressos em conjunto ou separados, tendo ainda a opção dum quadro de altos e baixos.

A estrutura do programa é a seguinte:

100 — subrotina para desenho dos eixos
300 — subrotina usada no cálculo do número de dias desde o dia em que nasceu
400 — subrotina usada na soma
500 — títulos dos gráficos
1000 — rotina introdutória das opções
2000 — rotina da 2.ª opção
3000 — rotina da 1.ª opção
4000 — rotina da 3.ª opção

```
  10 REM SPECTRUM BIORHYTMS @ P.
A.GARFIELD
  15 REM Adaptado da ZX COMPUTIN
G por Paulo Metelo
  20 GO TO 1000
  50 DATA 31,28+(1 AND (m>2 AND
INT (yy/4)=yy/4)),31,30,31,30,31
,31,30,31,30,31,31
 100 REM ***************
 110 PLOT x+4-xx*6,72
 120 IF x/6<>INT (x/6) THEN RETU
RN
 130 PLOT x+4-xx*6,73: LET date=
date+1
 140 RESTORE 50: FOR q=1 TO m: R
EAD c: NEXT q
 150 IF (m=2 AND (INT (yy/4)=yy/
4)) THEN LET c=29
 160 IF NOT (INT ((date-1)/7)=(d
ate-1)/7 AND date>1) THEN GO TO
190
 170 FOR w=72 TO 75: PLOT x+4-xx
*6,w: NEXT w
 180 IF (x-xx*6)/8<28 THEN PRINT
 AT 13,(x-xx*6)/8;date-1: PRINT
AT 14,(x-xx*6)/8;A$((m-1)*3+1 TO
 m*3)
 190 IF date>c AND NOT (INT (y/4
)=y/4 AND m=2) THEN LET m=m+1: L
ET date=1
 200 IF date>c+1 THEN LET m=m+1:
 LET date=1
 210 RETURN
 300 REM ***************
 310 RESTORE 50: LET mo=0: FOR s
=1 TO m-1: READ q: LET mo=mo+q:
NEXT s
 320 LET ye=(yy-1900)*365+INT ((
yy-1901)/4)
 330 LET t=d+mo+ye
 350 RETURN
 400 FOR a=0 TO 40 STEP b: PLOT
108+a,173-c: NEXT a: RETURN
 500 PRINT AT 3,0;b$((ww-1)*12+1
 TO (ww-1)*12+12): RETURN
1000 REM **************
1010 CLS
1020 LET A$="JANFEBMARAPRMAYJUNJ
ULAUGSEPOCTNOVDECJANFEV"
1030 LET b$="intelectual emocion
al   fisico      "
1040 PRINT "selecionar opcao"''"
1)intelectual  emocional  e
  fisico  ciclos  em graficos
  separados"''"2)todos os ciclos
 juntos num so   grafico"''"3)um
a lista dos pontos altos e    ba
ixos para um periodo de doze  me
ses"
1050 INPUT dd
1060 CLS
1070 INPUT "dia do mes em que na
sceu ";d
1080 INPUT "mes ";m
1090 INPUT "ano ";y: LET yy=y+(1
900 AND y<1900): GO SUB 300
1100 LET dob=t: INPUT "ENTER mes
 que quer calcular ";v: LET m=v
1110 INPUT "ano ";y: LET yy=y+(1
900 AND y<1900): LET u=1: LET d=
1: GO SUB 300
1120 LET new=t
1130 LET dif=new-dob
1140 LET xx=dif
1150 LET ww=1: IF dd=1 THEN GO T
O 3000
1160 IF dd=3 THEN GO TO 4000
2000 REM ***************
2010 PRINT AT 0,0;"intelectual="
: LET b=8: LET c=2: GO SUB 400
2020 PRINT "  emocional=": LET b
=2: LET c=10: GO SUB 400
2030 PRINT "     fisico=": LET b
=4: LET c=18: GO SUB 400
2040 LET date=1: LET xx=dif
2050 FOR x=xx*6 TO 250+xx*6
2060 LET y=72+60*SIN (x/99*PI)
2070 IF x/6=INT (x/6) THEN PLOT
x+4-xx*6,y
2080 LET y=72+60*SIN (x/99*PI/23
*33)
2090 IF x/2=INT (x/2) THEN PLOT
x+4-xx*6,y
2100 LET y=72+60*SIN (x/99*PI/28
*33)
2110 PLOT x+4-xx*6,y
2120 GO SUB 100
2130 NEXT x
2140 INPUT "enter P to print";v$
: IF v$="p" THEN COPY
2150 GO TO 1000: STOP
3000 REM ***************
3010 GO SUB 500
3020 LET date=1: LET xx=dif
3030 FOR x=xx*6 TO 250+xx*6
3040 LET y=72+60*SIN (x/99*PI)
3050 PLOT x+4-xx*6,y
3060 GO SUB 100
3070 NEXT x
3080 LET date=d: LET xx=dif: LET
 m=v
3090 INPUT "enter P to print";v$
: IF v$="p" THEN COPY : LPRINT :
 LPRINT : LPRINT : LPRINT
3100 CLS : LET ww=ww+1
3110 GO TO 3040+ww*40
3120 GO SUB 500
3130 FOR x=xx*6 TO 250+xx*6
3140 LET y=72+60*SIN (x/99*PI*33
/28)
3150 GO TO 3050
3160 GO SUB 500
3170 FOR x=xx*6 TO 250+xx*6
3180 LET y=72+60*SIN (x/99*PI*33
/23)
3190 GO TO 3050
3200 GO TO 1000: STOP
4000 REM ***************
4010 PRINT "     altos e baixos":
 PRINT " data     int   emoc  fi
si   sig": LET gg=0: LET date=d:
 FOR x=xx TO xx+365
4020 PRINT AT 2,0;A$(3*m-2 TO 3*
m)
```

```
4030 LET gg=0: LET date=d
4040 FOR x=xx TO xx+365
4050 LET aa=SIN (x*PI*2/33)
4060 LET bb=SIN (x*PI*2/28)
4070 LET cc=SIN (x*PI*2/23)
4080 RESTORE 50: FOR q=1 TO m: R
EAD c: NEXT q
4090 LET ch=0
4100 IF (m=2 AND (INT (yy/4)=yy/
4)) THEN LET c=29
4110 LET date=date+1
4120 IF date>31 AND m=12 THEN LE
T m=1: LET date=1: LET yy=yy+1:
LET ch=1
4130 IF date>c AND NOT (INT (y/4
)=y/4 AND m=2) THEN LET m=m+1: L
ET date=1: LET ch=1
4140 IF date>c+1 THEN LET m=m+1:
 LET date=1: LET ch=1
4150 IF (ch=1 AND gg<20) THEN PR
INT AT gg+2,0;A$(3*m-2 TO 3*m)
4160 IF (aa+bb+cc<2.4) AND (aa+b
b+cc>-2.4) THEN GO TO 4300
4170 LET gg=gg+1
4180 PRINT AT gg+1,0;date;TAB 2;
"-";m;"-";yy-1900;" ";
4190 IF aa<0 THEN PRINT TAB 10;"
-";
4200 PRINT TAB 11;ABS (INT (aa*1
00));
4210 IF bb<0 THEN PRINT TAB 16;"
-";
4220 PRINT TAB 17;ABS (INT (bb*1
00));
4230 IF cc<0 THEN PRINT TAB 22;"
-";
4240 PRINT TAB 23;ABS (INT (cc*1
00));
4250 IF cc<0 THEN PRINT TAB 29;"
-";
4260 LET ee=ABS (aa+bb+cc)-2.3:
LET ee=INT (ee/7*100+.5): PRINT
TAB 30;ee
4270 IF gg<20 THEN GO TO 4300
4280 INPUT "enter P to print";v$
: IF v$="p" THEN COPY
4290 LET gg=-2: CLS
4300 NEXT x
4310 INPUT "enter P to print";v$
: IF v$="p" THEN PRINT AT gg+2,0
;"  ": COPY
4320 GO TO 1000: STOP
9999 SAVE "BIORITMO" LINE 1: PRI
NT "VERIFY": VERIFY "BIORITMO":
CLS : PRINT "esta bem gravado"
```

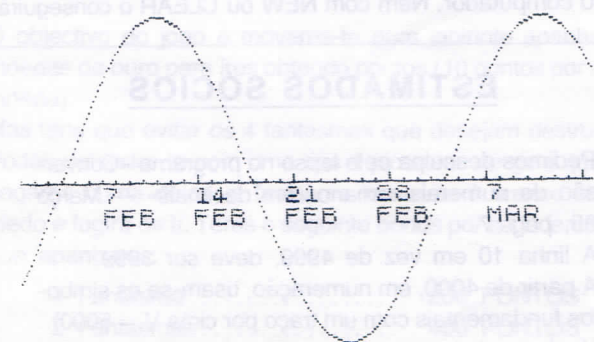
```
selecionar opcao

1)intelectual emocional e
  fisico ciclos em graficos
  separados

2)todos os ciclos juntos num so
  grafico

3)uma lista dos pontos altos e
  baixos para um periodo de doze
  meses
```

```
emocional

   7      14      21      28      7
  FEB    FEB     FEB     FEB     MAR
```

```
altos e baixos
data     int   emoc   fisi   sig
15-6-85  -100  -79    -82    -4
16-6-85  -98   -91    -95    -7
17-6-85  -91   -98    -100   -8
18-6-85  -82   -100   -98    -7
19-6-85  -70   -98    -89    -4
28-6-85   81    62     97     2
29-6-85   90    78     99     6
30-6-85   97    90     94     7
1 -7-85   99    97     81     7
2 -7-85   98    100    63     5
```

RESPOSTA AO DESAFIO

Fernando Duarte

Resposta ao desafio proposto na revista de Fevereiro sobre os números amigáveis.

```
  10 INPUT "Ate que valor quer e
ncontrar numeros amigaveis ?";a
  20 FOR i=2 TO a
  30 LET aa=1: LET b=i/2
  40 FOR j=2 TO b
  50 IF i/j=INT (i/j) THEN LET a
a=aa+j
  60 NEXT j
  80 IF aa>i THEN GO TO 200
  90 LET b=aa/2: LET bb=1
 100 FOR j=2 TO b
 110 IF aa/j=INT (aa/j) THEN LET
 bb=bb+j
 120 NEXT j
 130 IF bb<>i THEN GO TO 200
 140 PRINT i;" e ";aa;" sao nu
meros amigaveis"
 200 NEXT i
```

António Manuel de Bastos Pereira

```
   1 REM ======================
===========Programa elaborado por=
====================A. Bastos=======
==================================

   2 REM ======================
================ENTRADA DE DADOS====
==================================

  10 PRINT AT 0,6;"NUMEROS AMIGA
VEIS"''"Introduza o intervalo [a
,b] com a>=2 e b>=a, no qual pre
tende   pesquizar se existem ou
nao     numeros amigaveis."
  20 INPUT "a= ";a;" b= ";b
  30 IF a>=2 AND b>=a THEN GO TO
 50
  40 CLS : PRINT AT 10,0;"Dados
incorrectos - Prima ENTER": PAUS
E 0: RUN
  50 CLS : PRINT AT 10,0;"Progra
ma em processamento"
  60 REM ======================
============PROGRAMA PRINCIPAL====
==================================

  70 FOR j=a TO b
  80 LET i1=0: LET k1=j/2
  90 FOR i=1 TO k1
 100 IF j/i=INT (j/i) THEN LET i
1=i1+i
 110 NEXT i
 120 LET i2=0: LET k2=i1/2
 130 FOR i=1 TO k2
 140 IF i1/i=INT (i1/i) THEN LET
 i2=i2+i
 150 NEXT i
 160 IF j<>i2 THEN NEXT j: GO TO
 220
 165 REM ======================
================FIM / CONTINUAR====
==================================

 170 CLS : PRINT AT 10,0;"Os num
```

```
eros ";j;" e ";i1'"sao amigaveis
."
 180 PRINT ''"Pretende que cont
inue?  (s ou n)"
 190 IF INKEY$="s" OR INKEY$="S"
 THEN LET v=1: CLS : PRINT AT 10
,0;"Programa em processamento":
NEXT j: GO TO 220
 200 IF INKEY$="n" OR INKEY$="N"
 THEN STOP
 210 GO TO 190
 220 CLS : PRINT AT 10,0;"No int
ervalo dado nao existem numeros
amigaveis."
 230 IF v=1 THEN CLS : PRINT AT
10,0;"No intervalo dado nao exis
tem  mais numeros amigaveis." '
 240 PRINT ''"Nova tentativa? (
s ou n)"
 250 IF INKEY$="s" OR INKEY$="S"
 THEN RUN
 260 IF INKEY$="n" OR INKEY$="N"
 THEN STOP
 270 GO TO 250
```

PORMENORES

Adapt. e trad.
da «SINCLAIR USER»

Programação pode ser engraçada, mas muitas vezes desejas produzir um efeito no teu computador para o qual não consegues encontrar uma rotina. Muita gente experimenta estas dificuldades e por isso decidimos publicar uma série de pequenas rotinas. É difícil cobrir todas as áreas que compõem o mundo da programação. Nós vamos concentrar a nossa atenção especialmente nos aspectos principais da linguagem Basic.

PROGRAMAÇÃO ESTRUTURADA

Estrutura é a palavra mais snob da programação. Pessoas com máquinas muito caras, frequentemente ridicularizam o Spectrum porque o Basic «não suporta programação estruturada». Programação estruturada é um conceito, e os princípios podem ser usados em qualquer máquina. A ideia a seguires é a de cortares o teu programa numa série de pequenos blocos, de preferência subrotinas. Por exemplo, um jogo de arcádia pode ter os seguites blocos:

1000 Refere-se a ecran (screen) e variáveis
2000 Joysticks/teclado
3000 Movimentos
4000 Fim do jogo
5000 Definição de gráficos (UDG)

O resto do programa é apenas uma questão de ligar as rotinas entre si.

```
10 GOSUB 5000:GOSUB 100
20 GOSUB 2000
30 GOSUB 3000
40 se é o fim do jogo THEN GOSUB 4000
50 GO TO 20
```

Uma estrutura como esta torna muito mais fácil alterar o programa ou juntar-lhe outras características.

BONS HÁBITOS

Tenta usar nomes para as variáveis que indiquem o objectivo da variável.
Variáveis como «hor» e «ver» para coordenadas horizontais e verticais são mais fáceis de seguir do que o uso de «x1» e «x2»,que não tem qualquer significado.
O Basic foi feito para seguir o Inglês sempre que possível, portanto todos os antecedentes indicam que devemos continuar a seguir esta tradição.
Não cometas o erro de usar apenas uma letra para as variáveis, para poupar espaço. Se estás a fazer um programa, claro que o espaço é importante, mas precisarás de nomes que sejam fáceis de entender. No fim podes mudar as variáveis para nomes simples com 1 ou 2 caracteres, se for necessário. Usa muitas linhas REM para te lembrarem o que é que um determinado bloco do programa faz. Poderás saber tudo acerca do programa no dia em que o estás a fazer, mas é muito fácil esqueceres-te se estiveres uma semana longe da máquina.

SCROLL INFINITO

Pode ser frustrante, especialmente durante um jogo, se o programa parar continuamente e perguntar se quer Scroll ou não. O Scroll é controlado por uma das variáveis do sistema que conta o número de páginas, pedindo para accionar qualquer tecla antes de continuar ou N para parar. A linha 10 POKE 23692,255 inutiliza o sistema. Se queres ter a certeza de que nunca precisarás de premir uma tecla para fazer Scroll, então verifica se a linha está repetida em diversos ponto do programa, por exemplo incorporada num loop.

ARMAZENAR GRÁFICOS

Os caracteres UDGs estão guardados numa parte especial da memória com início no endereço 32600. Cada caractere está guardado com 8 bytes consecutivos feitos com 8 byts cada um, os quais representam 1 pixel (ponto no ecran). Quando postos no ecran os 8 bytes formam uma grelha. Usando UDGs standard temos apenas memória suficiente para as letras de A a U. Estas estão colocadas na RAM e podem ser alteradas usando as instruções estruturadas à volta da fórmula geral POKE USR «caractere gráfico» + número de byte (1 a 8). Por exemplo, o sétimo byte da letra A entraria usando a instrução POKE USR «A» + 7.
Como a área UDG está protegida, os gráficos que se situam dentro dela não podem ser destruídos, a menos que se desligue o computador. Nem com NEW ou CLEAR o conseguirá.

ESTIMADOS SÓCIOS

Pedimos desculpa pelo lapso no programa «Conversão de numerais romanos em decimais» — Março 85, pág. 7.
A linha 10 em vez de 4999, deve ser 3999.
A partir de 4000, em numeração, usam-se os símbolos fundamentais com um traço por cima V = 5000).
Nenhum símbolo pode ser usado mais de 3 vezes.

NOVOS PROGRAMAS

MUTANT MONTY

Como toda a gente MONTY tem duas ambições na vida: enriquecer rapidamente e ser um herói.
Aqui está a sua oportunidade apenas percorrendo as 40 salas e recolhendo todo o ouro para no final salvar a donzela. Tudo isto com apenas 5 vidas e lutando contra o relógio (o resultado é em função do tempo que gastou) pois se não recolher o ouro a tempo perde uma vida.
No menu inicial carregue em «1» para jogar com as seguintes teclas (ou escolha joystick com «2» ou «3»):

TECLAS A USAR

1 — suspender o jogo
2 — música on/off
CAPS ou X — esquerda
Z ou C — direita
Q a P — para cima
A a ENTER — para baixo
CAPS + SPACE — abortar o jogo

SPACE SHUTTLE II

Esta é a tua centésima primeira missão no Sistema de Transporte Espacial. Estás na direcção da nave. O teu alvo encontra-se a aproximadamente, 210 milhas náuticas acima da terra: um satélite em órbita com problemas no giroscópio, intencionalmente programados.
A tua missão é ir ao encontro ao satélite o máximo de vezes que puderes, usando o mínimo de gasolina; e depois voltares a salvo para a terra. Sempre que o conseguires encontrar com sucesso, o satélite está programado para ficar cada vez mais errante.
Isto é um teste às tuas capacidades como piloto. Serás avaliado no fim de cada voo.

FALL GUY

Este jogo pretende funcionar como uma cena para o cinema. O objectivo é saltares do cimo do tunel para cima do comboio percorreres os telhados das carruagens com o comboio em andamento. Tens que ter o máximo cuidado para não caires entre as carruagens.
Se conseguires executar esta cena sem problemas aumentarás o capital do director do filme e, assim, talvez ele te deixe tirar umas boas férias.

HAUNTED HEDGES

O objectivo do jogo é moveres-te num labirinto apanhando moedas de ouro para ires obtendo pontos (10 pontos por cada moeda).
Mas tens que evitar os 4 fantasmas que desejam destruir-te. Podes defender-te com uma das 4 picaretas de gelo quando pegares numa delas, o fantasma tornar-se-á branco com o medo e fugirá de ti. Terás o seguinte bónus por cada fantasma que apanhares:

1 Fantasma	200 PONTOS
2 Fantasmas	400 PONTOS
3 Fantasmas	800 PONTOS
4 Fantasmas	1 600 PONTOS

Durante o jogo aparecerte-ão mais tesouros pelos quais terás um bónus extra:

Arca do tesouro	200 PONTOS
Safira	500 PONTOS
Talismã	1 000 PONTOS

Quando apanhares todas as moedas e tiveres usado todas as picaretas, passarás para outro nível onde existem mais moedas e onde os fantasmas se movem mais depressa. Depois de obteres 10000 ganharás uma vida extra.
Diverte-te.

THE DUKES OF HAZZARD (Elite)

O velho vilão Boss Hogg está novamente a preparar das suas. Os Dukes devem-lhe $5000, que lhes foram emprestados para eles comprarem algumas máquinas agrícolas.
O Boss Hogg quer o dinheiro de volta e avisou os Dukes que se eles não pagassem o que lhe deviam em 72 horas, ele ficaria com o General Lee como garantia.
O Bo e o Luke decidiram entrar na corrida de automóveis de Hazzard — primeiro prémio $5000 — que começava no dia seguinte.
Normalmente os Dukes teriam bastantes possibilidades de ganhar a corrida, mas com toda a polícia de Hazzard e o Boss Hogg atrás deles para os impedirem de alcançar os seus fins, os Dukes não estão tão certos disso!
O objectivo do jogo é chegar ao princípio da corrida de automóveis de Hazzard, conduzindo o General Lee desde a casa dos Dukes e por uma estrada extremamente acidentada. Ao longo do teu percurso tens que manter-te à frente do Roscoe, cuja maior ambição é prender os Dukes.
Esta pode ser a oportunidade de Roscoe tornar o seu sonho realidade, pois tem a ajuda de toda a polícia de Hazzard. Tem cuidado com O Boss que está na sua carruagem, na estação.
Toma cuidado, também com Daisy — aborrece-a e pagarás bem caro.
Os Dukes têm uma série de truques na manga: dinamite, e quando o big-bang começa um espectacular salto do General Lee, poderá resolver o problema.
Diverte-te.

TOWER OF DESPAIR

É um jogo de aventuras.
Uma terrível guerra entre as forças do mal e as forças do bem, devastara Aelandor. Com a vitória das forças do bem Aelandor reencontrará a paz.
Muitos anos passaram desde essa terrível guerra.
Agora és o guerreiro Mage do castelo Argent, filho do herói Thorvald. Viveste sempre em paz e em prosperidade — parecem séculos desde a terrível guerra em que o teu pai e Elwood derrotaram Malnor, o senhor do mal. Nessa altura tu eras uma criança mas nunca te cansas de ouvir a história.
Numa manhã de sol quando trabalhavas no santuário do castelo Argent, o globo de mensagens dos feiticeiros de

Aelandor começou a emitir terríveis mensagens. O Demónio voltara a despertar. Tudo fora planeado na torre do desespero e desta vez o seu poder é superior. Tu como guerreiro mais destemido tens que lutar contra este terrível poder. A tua tarefa será difícil e extremamente perigosa.
Mas o destino de Aelandor está nas tuas mãos, por isso não podes falhar.
Boa sorte!

3-D STARSTRIKE (Realtime Software)

O objectivo deste jogo é destruir os inimigos que sucessivamente vão surgindo no ecran ao longo dos seus 5 quadros. Não nos podemos deixar atingir, pois cada vez que uma nave choca connosco, vamos perdendo energia assim como a nossa condição de segurança vai passando de verde para amarela e vermelha, consecutivamente.
Se ao chegarmos ao quinto quadro tivermos energia suficiente e conseguirmos atravessar o túnel, voltamos ao início do jogo, num nível de dificuldade inferior.
O programa é compatível com Joysticks ou teclado. Para a opção de teclado, as teclas a utilizar são:

Q — Subir　　I Esquerda
A — Descer　　P — Direita
M — Disparar

* O programa tem 5 níveis de dificuldade. Para os escolher, basta carregar numa tecla de 0 a 4.

FINDERS KEEPERS

Trata-se de um jogo de Arcádia.
A história começa quando o guerreiro mágico quer tornar-se um membro da tábua poligonal. O rei manda-o para o castelo de SPRITELAND, para ele provar o seu valor. Mas o mágico entretanto mudou de ideias . . .
O mágico tem, portanto 2 hipóteses:
1) Fugir do castelo de SPRITELAND
2) Recolher o máximo de tesouros possível para agradar ao rei. Para isso tem que recolher os objectos que encontrar e evitar os monstros que lhe diminuem a energia.
O programa contém instruções mais pormenorizadas, assim como permite a redefinição das teclas.

HALAGA (Interceptor Micro's)

Contratado pela federação de Pesquisa Espacial para examinar o novo sistema de Cygnus Major de recursos minerais, encontrarás uma recepção hostil. Centenas de insectos, como criaturas vindas de outra dimensão, tentarão destruir a tua missão.
Apenas a tua perícia e o teu laser poderão ajudar-te.

Teclas: CAPS SHIFT — Esquerda
Z — Direita
SPACE — Fire

VALKYRIE 17

Uma noite chegas a casa e descobres uma mensagem no teu gravador de mensagens.
Algo que julgavas desaparecido para sempre, levantou a cabeça monstruosa novamente. Valkyrie 17 está activa.
Durante as 5 noites seguintes, uma série de telefonemas ansiosos, convencem-te de que o assunto merece uma investigação. Recebeste um dossier das actividades de Valkyre 17. Juntos, diversos fragmentos de relatórios feitos nos últimos 40 anos fazem-te, lentamente, começar a compreender tudo. Drakenfeur, Heinrich e Reichsmuller . . . O emblema pressionado contra a tua mão na estação. E aquele telefonema desesperado pedindo socorro, do Hotel Glitz. As tuas investigações no Hotel Glitz parecem não te levar a lado nenhum. Mas quando te dirigias para o bar sentes algo mover-se na sombra, uma pancada na cabeça e tudo fica negro . . .
Valkyrie 17 é uma aventura, que reune excelentes gráficos com textos que te dão as diferentes localizações. Vais encontrar uma série de personagens. Serás ajudado por algumas, enquanto outras desejam apenas a tua morte.
Do outro lado da cassete tens as mensagens telefónicas e o dossier das actividades de Valkyrie 17.
Tem cuidado e boa sorte pois vais precisar dela.

BATTLE CARS

É um jogo de perícia e estratégia podendo estas ser utilizadas de diferentes maneiras. Existem 3 diferentes arenas para a batalha — um circuito, o centro de uma cidade e o autódromo (um estádio vazio onde as tuas armas e a tua perícia serão testadas).
Como jogador, poderás criar as tuas manobras e testá-las, correndo no circuito contra um automóvel controlado pelo computador.
Poderás também testar as tuas capacidades no autódromo, num combate contra o computador.
Mas o último teste à tua perícia será quando competires com um jogador, numa luta até à morte, em qualquer dos circuitos.
No jogo Battle Cars podes escolher os teus carros e enchê-los com as armas que pretenderes.
Boa sorte!

747 FLIGHT SIMULATOR

Este programa produz uma simulação das características de voo de um Jumbo dando uma cuidada visão do exterior e um retrato fiel dos painéis e instrumentos de voo. O teclado (ou Joystick) é usado para controlar o avião. Todos os instrumentos essenciais de voo estão representados no ecran, e estes conjuntamente com outras indicações, manter-te-ão informado da situação, condição e execução do teu voo.

UTILITÁRIOS

MEGA BASIC (550$00)

É um programa em código máquina (cerca de 22K) com a finalidade de aumentar a capacidade do ZX Spectrum, dando-lhe características de um computador de preço 5 vezes superior ao seu.
(Janelas, diferentes tipos de caracteres, assim como uma série de novos comandos).
O teclado funcionará como um computador normal em que todos os comandos e funções serão dados ao computador letra a letra. O método de EDIT de uma linha será conseguido pelo cursor que se move no ecran independentemente do habitual cursor do INPUT. O ecran pode ser dividido em 3 sec-

ções: secção de INPUT de comandos e display de mensagem de erros; secção de OUTPUT do programa; secção de listagem automática. (Estas áreas são definidas pelo utilizador). Os três tipos diferentes de caracteres estão definidos para:

— 64 colunas/24 linhas (metade do tamanho normal)
— 32 colunas/24 linhas (caracteres normais)
— 32 colunas/12 linhas (dobro da altura)

com a possibilidade de 5 formas diferentes de display dos mesmos.
Também a posição e o tamanho das janelas podem ser totalmente definidas pelo utilizador, obtendo várias janelas simultâneas se o desejar. Cada janela pode ser limpa individualmente, invertida, com scroll para cima, esquerda e direita. Pela primeira vez o Spectrum terá as suas próprias rotinas de sprites. Tudo o que tem a fazer é definir a direcção que pretende para o movimento da sprite podendo definir também uma forma de reconhecer se uma sprite colide com um outro gráfico.

Breve explicação dos 25 novos comandos MEGA BASIC

AUTO — Produção automática do número de linha
BACKUP — Cópia de Files
CHANGE — Manipulação da File
CLW — Limpa uma janela no ecran
CURRENT — Troca a janela usada
DELETE — Limpa um bloco de Basic
DOWN — PRINT de uma string no fundo do ecran
EDIT — Apresenta uma linha de programa para EDIT
EXAMINE — Display de headers das Files
FADE — Efeitos especiais no ecran
FONT — Seleccionar o tipo de caracteres a usar
FX — Funções diversas
INVERT — Altera INK para PAPER e vice-versa
KEY — Definir novas teclas
MODE — Altera o tamanho dos caracteres em determinado momento
MON — Salto para o painel principal
PAN — Scroll da janela pixel/pixel à esquerda ou direita
PLAY — Efeitos sonoros
RESTART — Equivalente a «ON ERROR GOTO»
SPEED — Com TRON reduz a velocidade de execução do programa
SWAP — Altera atributos
TROFF — Desligar o mecanismo de TRACE
TRON — Ligar o mecanismo de TRACE
VDU — Equivalente a PRINT CHR$
WINDOW — Define localização e tamanho da janela no ecran

TASWIDE — Spectrum 16K/48K

Programa totalmente em código máquina com 1495 bytes, permite utilizar 64 caracteres por linha no ecran, podendo ser utilizado também em conjunto com programas requeridos por alguns Interfaces para saída na impressora.

COMPILER FP — **Spectrum 16K/48K**

Compilador Basic. Tem como função, traduzir programas Basic em código máquina, aumentando assim a velocidade de execução de um seu programa em cerca de 10X a velocidade normal. Antes de dar início à compilação de qualquer programa, poderá definir a sua posição em memória e o FP COMPILER em seguida traduzirá quase na totalidade todos os comandos funções Basic.

EDITOR/ASSEMBLER 16K/48K (Picturesque) — **Spectrum**

MONITOR/DISASSEMBLER 16K/48K (Picturesque) — **Spectrum**

(Deslocáveis na memória e com a possibilidade de trabalharem conjuntamente).

TASCOPY (500$00)

Cópia de ecrans (screens), via ZX Interface 1

Contém 2 programas:

1 — Para reprodução de um ecran monocromo, por linhas horizontais
2 — Para reprodução de ecrans em grande escala, por linhas verticais, utilizando pontos com diferentes densidades, de forma a se poder distinguir as cores utilizadas.

Sinclair ZX Spectrum
Sinclair ZX Interface 1
E as impressoras
Epson FX-80
Epson RX-80
Epson MX-80 type III
Shinwa CP-80
Mannesman Tally MT 80
Star DMP 510/515
Brother HR5

TASPRINT (500$00)

Com aplicação no programa «Tasword Two».
Permite a impressão de vários tipos de caracteres com a altura 2x superior à normal.
Sublinhados, escrita invertida (inverse print) e outras formas de destacar palavras ou frases de um texto.

TASMERGE (500$00)

Torna possível transferir do programa Masterfile, files contendo até 15 campos, para o Tasword Two, podendo então sairem na impressora segundo os seus caracteres e formas. (com aplicação apenas em microdive)

SUPERCODE III (version 3.5) (600$00)

Um excelente toolkit, compatível com Spectrum Plus e Spectrum 48K.
Consiste em 152 rotinas em código máquina.
— Dá mais flexibilidade aos seus programas. (Inclui rotinas como: TRACE, ON ERROR GOTO, ON BREAK GOTO, etc. . . .).
— Efeitos especiais: todo o tipo de Scroll (esquerda, direita, para cima, para baixo e diagonal), troca instantânea de cor (PAPER, INK, BRIGHT e FLASH), MERGE e INVERSE de ecrans (screens), efeitos sonoros, etc. . . .
— Compatível com todos os acessórios (Sinclair) tais como: ZX Printer, Prism Modern, Interfaces 1 e 2, etc. . . .
— O programa em Basic, permite o acesso às 152 rotinas, não tendo no entanto qualquer influência na sua execução, sendo a sua função essencial a de apoio como instruções (200 pág. de ecran).

CLUBE Z80

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z80** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.

2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).

3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.

4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME ....

IDADE .... COMPUTADOR TIPO ....

PROFISSÃO ....

ENDEREÇO ....

.... TELEF. ....

ASSINATURA ANUAL — Esc. 1 500$00 ☐

ASSINATURA SEMESTRAL — Esc. 750$00 ☐

CHEQUE OU VALE DO CORREIO

N.º ....

BANCO ....

DATA ....../....../......

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de .... (inclusive)

MERCADO Z80

O MERCADO Z80 É UMA SECÇÃO DO CLUBE Z80 QUE EMPRESA AOS SEUS SÓCIOS PROGRAMAS E LIVROS (SPECTRUM) PARA CONSULTA E MELHOR CONHECIMENTO/APROVEITAMENO DE MICROCOMPUTADORES.

— PROGRAMAS: Todos os jogos e utilitários que existem no CLUBE Z80, excepto programas de cópia e programas com direitos de autor (Ex.: "Cálculo de Estruturas").

— LIVROS: Cerca de 40 títulos diferentes.

COMO TORNAR-SE SÓCIO DO MERCADO Z80?

Para poder ter em sua posse 5 cassetes ou livros durante um mês, basta enviar-nos um depósito de Esc. 2000$00 (garantia de que os materiais nos serão devolvidos em estado de conservação e funcionamento idêntico àquele em que foram enviados).
Ao mesmo tempo, deverá remeter-nos a quantia de Esc. 1000$00 que será a base da sua "Conta-Corrente". Essa quantia servirá para pagar as suas despesas:

— Taxa de utilização dos produtos: 250$00 (referente a 5 unidades, entre livros e programas).

— Instruções dos programas (no caso de o sócio não as devolver, debitar-lhe-emos 5$00 por folha).

— Embalagem Postal: 20$00 a 30$00 (no caso de o pedido ser feito via CTT.

— Portes dos CTT's: 40$00 a 80$00 (no caso de o pedido ser feito via CTT.

IMPORTANTE!

— O depósito de 2000$00 pertence integralmente ao sócio desde que os materiais por ele utilizados nos sejam devolvidos nas mesmas condições em que saíram do CLUBE Z80. Assim, quando o sócio desistir do MERCADO Z80, essa quantia ser-lhe-á entregue.
Em caso de extravio, danos ou avarias dos materiais, o sócio pagará o valor comercial dos respectivos produtos (a descontar no depósito de 2000$00).

— No caso de os produtos seguirem via CTT, o sócio não pagará para levantar a encomenda. As despesas serão pagas por nós, no momento da expedição, e debitadas ao sócio (a descontar no depósito de 1000$00).

— Quando as suas despesas estiverem a atingir os 1000$00 avisá-lo-emos, e o sócio deverá renovar essa quantia de modo a cobrir despesas seguintes.

— A taxa de utilização dos produtos é fixa — 250$00. Ela refere-se ao conjunto de 5 unidades. (Pagará sempre 250$00 mesmo que peça só uma unidade).

QUE PRODUTO E QUE QUANTIDADES?

O sócio nunca pode pedir mais do que 5 unidades de cada vez (entre livros e cassetes). Quanto a livros não poderemos empresar mais do que um. Assim, o sócio poderá pedir:

— 5 cassettes
ou
— 4 cassettes + 1 livro

QUAL O TEMPO DE UTILIZAÇÃO?

O sócio poderá ficar com os produtos durante 1 MÊS, no máximo. Findo esse período, deverá devolvê-los ao CLUBE Z80.
O MERCADO Z80 só atenderá dois pedidos por mês, para cada sócio.

COMO FAZER O PEDIDO?

Numa carta, escreva pelo menos 10 títulos (por ordem de prioridade). Se os 5 primeiros não estiverem disponíveis, enviaremos os outros evitando grandes esperas de produtos que estejam em circulação.
Ao devolver os produtos, inclua uma carta com o pedido seguinte.

IMPORTANTE! O SÓCIO SÓ PODERÁ EFECTUAR UM NOVO PEDIDO JUNTAMENTE COM A DEVOLUÇÃO DO MATERIAL CORRESPONDENTE AO PEDIDO ANTERIOR (ou depois, se preferir).

Se estiver interessado no MERCADO Z80, faça já o seu 1.º pedido, enviando 3000$00 e o cupão abaixo devidamente preenchido.

INSCRIÇÃO NO MERCADO Z80

NOME ______________________

ENDEREÇO ______________________

CÓDIGO POSTAL ______________________

TELEFONE ____________

ENVIO 3 000$00 (2 000$00 como garantia de que devolverei os produtos em boas condições + 1 000$00 para a minha «Conta Corrente» em:

☐ Cheque n.º ____________ ☐ Vale Postal ☐ Dinheiro

Banco ____________ N.º ____________ ____________

Data ___/___/___ *Assinatura do Sócio* ______________________

É SÓCIO DO CLUBE Z80? ☐ SIM ☐ NÃO